

Europäisches Patentamt  
European Patent Office  
Office européen des brevets



(11) EP 0 921 473 A2

(12) EUROPEAN PATENT APPLICATION

(43) Date of publication:  
09.06.1999 Bulletin 1999/23

(51) Int. Cl.<sup>6</sup>: G06F 13/38, G06F 13/12

(21) Application number: 98309858.3

(22) Date of filing: 02.12.1998

(84) Designated Contracting States:  
AT BE CH CY DE DK ES FI FR GB GR IE IT LI LU  
MC NL PT SE  
Designated Extension States:  
AL LT LV MK RO SI

(72) Inventor: Mori, Junichi  
Ohta-ku, Tokyo (JP)

(74) Representative:  
Beresford, Keith Denis Lewis et al  
BERESFORD & Co.  
2-5 Warwick Court  
High Holborn  
London WC1R 5DJ (GB)

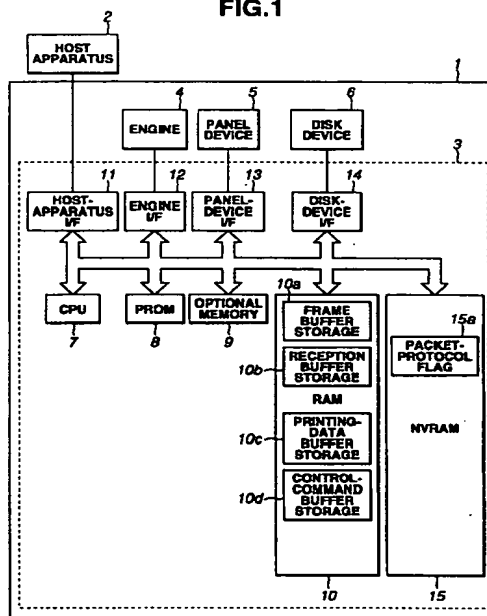
(30) Priority: 05.12.1997 JP 33567997

(71) Applicant:  
CANON KABUSHIKI KAISHA  
Tokyo (JP)

(54) Interface control apparatus

(57) In two-way communication with a host apparatus, when packet data is received, a packet protocol is used, and when nonpacket data is received, a non-packet protocol is used. Control is performed so that the destination of each data in the received data is switched between a printing-data buffer storage and a control-command buffer storage depending on whether or not the packet protocol is to be used.

FIG.1



EP 0 921 473 A2

## Description

### BACKGROUND OF THE INVENTION

#### Field of the Invention

[0001] The present invention relates to a communication control apparatus for performing two-way communication processing with a data processing apparatus via a predetermined interface, a method for processing data of the communication control apparatus, and a storage medium storing programs capable of being read by a computer.

#### Description of the Related Art

[0002] Conventionally, data communication between a host apparatus and an image input/output apparatus, such as a printer, a scanner or the like, is executed via a predetermined interface. A Centronics interface, serving as an interface of this type, is a main interface for transferring data from a host computer to a printer, and has the feature that inexpensive high-speed data transfer can be performed with standards for transmitting data from a computer to a printer which have been developed by Centronics Corporation of U.S. A. for its own printers. Although the Centronics interface is widely used as means for transmitting data from a personal computer (PC) to a printer, it is not standardized.

[0003] Although the Centronics interface has many variations depending on each combination of a host computer and a printer, the variations are basically similar. Basic controls are performed using three signal lines, i.e., DATA STROBE, ACK and BUSY. A transfer method using these three signal lines is called a compatibility mode in contrast to a nibble mode, a byte mode, an ECP (extended capabilities port) mode and the like (to be described later).

[0004] The compatibility mode will now be described in detail with reference to drawings. It is impossible to describe all of many variations which exist in the compatibility mode as described above. Hence, a description will be provided illustrating a laser-beam printer made by Canon Inc.

[0005] FIG. 22 is a plan view illustrating the pin arrangement of a printer interface in a printing system to which a communication control apparatus of this type can be applied. In FIG. 22, 36 pins are arranged.

[0006] Numerals in FIG. 22 represent the numbers of respective pins, each of which corresponds to one signal line.

[0007] In FIG. 22, a pin 1 is a signal line for DATA STROBE, and usually assumes a "High" state. When the state shifts to a "Low" state, a printer reads the states of pins 2 - 9 for data lines DATA1 - DATA8, respectively.

[0008] The data lines DATA1 - DATA8 represent information for the 0th - 7th bits, respectively, of data trans-

mitted from a host apparatus. On each of these data lines, data is "1" and "0" in the "High" and "Low" states, respectively. In a stationary state, data is indefinite. Data is effective only when the DATA STROBE line assumes the "Low" state.

[0009] A pin 10 is for an acknowledge line ACK, and assumes the "High" state in a stationary state. A pulse for providing the "Low" state is generated when the state shifts from the stationary state.

[0010] A pin 11 is for a busy line BUSY, on which a state signal indicating whether or not the printer can receive data from the host apparatus is provided. On this line, the "Low" state represents a state in which a signal can be received, and the "High" state represents a state in which a signal cannot be received.

[0011] A pin 12 is for a printer error line PE, which assumes the "High" state when an error is generated in the printer, and assumes the "Low" state in other states. A pin 13 is for a select line SELECT, which assumes the "High" state when the printer is in an on-line state, and assumes the "Low" state in other states. A pin 14 is for a signal line AUTO FD, which is unused.

[0012] A pin 15 is for a signal line AUXOUT1, which always assumes the "High" state. The ground level of the printer is applied to a pin 16. A pin 17 is for a frame ground line Frame GND, which corresponds to the frame ground of the printer.

[0013] A voltage of +5 V is applied to a pin 18, which always assumes the "High" state. Pins 19 - 30 are connected to the ground. A pin 31 is for an input prime line INIT, which always assumes the "High" state. By making the input prime line INIT to the "Low" state, input prime processing is performed.

[0014] A pin 32 is for a signal line FAULT, which assumes the "High" state when the printer is in an on-line state, and assumes the "Low" state in other states. A pin 33 is for a signal line AUXOUT2, which always assumes the "Low" state. A pin 34 is for a signal line AUXOUT3, which always assumes the "Low" state. A pin 35 is for a signal line AUXOUT4, which always assumes the "High" state. A pin 36 is for a signal line SELECTIN, which is unused.

[0015] In the compatibility mode, the pins 1 - 11 for DATA STROBE, DATA1, DATA2, DATA3, DATA4, DATA5, DATA6, DATA7, DATA8, ACK and BUSY, respectively, are mainly used. FIG. 23 illustrates handshaking in the compatibility mode.

[0016] FIG. 23 illustrates a timing chart for explaining the state of data transfer processing in the printing system. The timing chart shown in FIG. 23 corresponds to handshaking in the compatibility mode. In this case, 1-byte data is transmitted.

[0017] The compatibility mode can only perform data transfer in one direction from the host apparatus to the printer, and cannot perform data transfer from the printer to the host apparatus.

[0018] In order to solve this problem, two-way Centronics communication handshaking is standardized by the

IEEE (Institute of Electrical and Electronics Engineers) as a superior alternative of the compatibility mode (IEEE 1284-1994).

[0019] In the IEEE 1284-1994, a plurality of communication modes, such as the nibble mode, the byte mode, the ECP mode and the like, are newly provided in addition to the above-described compatibility mode (the conventional handshaking for transferring data from a host apparatus to a printer).

[0020] Connectors and cables having the same shape as in the compatibility mode can be used in all of the newly added modes. The nibble mode is for transferring data from a printer to a host apparatus. By alternately using the nibble mode and the compatibility mode, two-way communication between a host apparatus and a printer can be realized.

[0021] That is, by performing transmission from a host apparatus to a printer in the compatibility mode and performing transmission from the printer to the host apparatus in the nibble mode, two-way communication can be realized.

[0022] In the nibble mode, control is performed using the acknowledge signal line ACK and the signal line AUTO FD, and data is set on the four signal lines, i.e., BUSY, PE, SELECT and FAULT.

[0023] 8-bit (1-byte) transfer is realized by dividing 1 byte of data into two 4-bit portions, and first transmitting lower-order 4 bits followed by transmission of higher-order 4 bits. The signal line FAULT is also used for indicating by the printer whether or not data to be transmitted to the host apparatus is ready at a specific timing. Handshaking is performed without using the signal lines DATA1, DATA2, . . . , DATA8 which are controlled only by the host apparatus in the compatibility mode. These signal lines DATA1, DATA2, . . . , DATA8 are called data buses. Since the data buses are not used in the nibble mode, the host apparatus side need not provide hardware for dealing with data transmitted through the data buses. Hence, mounting can be relatively easily realized.

[0024] The byte mode is also for performing communication from a printer to a host apparatus, as the nibble mode. By alternately using the compatibility mode and the byte mode, two-way communication between the host apparatus and the printer can be realized.

[0025] That is, by performing transmission from the host apparatus to the printer in the compatibility mode, and performing transmission from the printer to the host apparatus in the byte mode, two-way communication can be realized. In the byte mode, control is performed using the signal lines STROBE, ACK, BUSY, PE, AUTO FD and FAULT, and data is set on the data buses (the signal lines DATA1, DATA2, . . . , DATA 8). This mode is more efficient than the nibble mode because 1-byte (8-bit) data is simultaneously transmitted. However, since the data buses are controlled by the printer side, the host apparatus side must provide hardware for dealing with transmitted data.

[0026] The ECP mode is for performing two-way communication between a host apparatus and a printer, and vice versa. By performing switching between Forward Phase (from the host apparatus to the printer) and Reverse Phase (from the printer to the host apparatus), two-way data communication can be realized in the same ECP mode.

[0027] In the ECP mode, control is performed using the signal lines STROBE, ACK, BUSY, AUTO FD, INIT, FAULT and PE, and data is set on the data buses.

[0028] In this case, since the data buses are also controlled by the printer side, the host apparatus side must provide hardware for dealing with transmitted data. The ECP mode has the features that overhead for shifting between the modes is small, and a higher transfer speed can be provided in handshaking for transfer in the Forward Phase than in handshaking in the compatibility mode. The outline of the IEEE 1284 has been described.

[0029] Data transmitted from the host apparatus to the printer via the Centronics interface is interpreted only by an interpretation portion of the printer side, and the printer performs printing processing in accordance with the result of interpretation. During transmission of a series of data, it is impossible to transmit other data, control commands and the like until the transmission of the series of data is completed.

[0030] That is, since a command for asking the state of the printer cannot be transmitted to the printer until the transmission of a series of data is completed even if the host apparatus side intends to know the state of the printer during the transmission, the host apparatus side cannot know the state of the printer.

[0031] In order to solve such a problem which will occur when using an I/F (interface) having only a single data path as the Centronics interface, it is now intended to standardize packet protocols.

[0032] For example, IEEE 1284.4 is one of packet protocol standards. The IEEE 1284.4 assumes various serial or parallel I/Fs, such as RS232C, USB (Universal Serial Bus), IEEE1394 and the like, in addition to the IEEE 1284.4 as physical I/Fs. An outline of the IEEE 1284.4 will be described later.

[0033] The packet protocol is a protocol in which, for example, a series of data is divided into a plurality of blocks, and the data is provided in the form of a packet by adding a header indicating packet information, such as the data size, channel IDs and the like, to the head of each block, so as to logically realize a plurality of channels and allow independent information exchange without causing interference between the channels.

[0034] By using such a packet protocol, it is logically possible to exchange a plurality of independent information (data, control commands and the like) in parallel on different channels. For example, it is possible to transmit a command asking the state of the printer during transmission of a series of data without awaiting the completion of the transmission.

[0035] By preparing a reception buffer storage for each channel, even if the reception buffer storage in a certain channel is fully occupied, other channels can be used without being influenced by that channel.

[0036] For example, when it is intended to transmit 20-Kbyte (kilobyte, hereinafter abbreviated as "KB" (1 KB = 1024 bytes)) data as printing data and 100-byte (hereinafter abbreviated as "B") data as a control command, a protocol other than a packet protocol can select only one of two methods, i.e., a method of first completing the transmission of the 20-KB printing data followed by the transmission of the 100-B control command, and a method of first completing the transmission of the 100-B control command followed by the transmission of the 20-KB printing data.

[0037] To the contrary, in a packet protocol, it is possible to transmit a control command during transmission of printing data. For example, the 20-KB printing data is divided into 10 blocks, and the 100-B control command is transmitted after completing transmission of a 2-KB block and before transmitting the next 2-KB block. Hence, a packet protocol is useful, for example, in real-time control of a printer by a host apparatus.

[0038] An outline of the IEEE 1284.4, an example of division of data into packets, and an example of transmission of printing data and a control command will now be described.

[0039] It can be said that the IEEE 1284.4 mainly provides two features, i.e., flow control between two apparatuses (for example, a host apparatus and a printing apparatus) utilizing a concept of credit and using data transfer conforming to the above-described IEEE 1284 standards (or any other appropriate interface), and multiplexing (provision of multichannels) for simultaneously exchanging a plurality of information (for example, data and control information).

[0040] In internal processing in the host apparatus and the printing apparatus, control by the IEEE 1284.4 is performed in an intermediate portion between processing of performing printing using transferred data (generally called "application") and the IEEE 1284, serving as actual data transfer portion. FIG. 24 illustrates the concept of such relationship.

[0041] FIG. 24 is a diagram illustrating the concept of relationship between each communication control program and application in the printing system.

[0042] As shown in FIG. 24, a control program according to the IEEE 1284.4 is present between the host apparatus and the printing apparatus which can communicate with each other via an interface cable, so as to perform communication between application and a control program according to the IEEE 1284.

[0043] Next, the configuration of data dealt with in the IEEE 1284.4 will be described. Data dealt with in the IEEE 1284.4 is always formed in units of a group comprising some bytes, which is called a packet. The configuration of the packet is defined as shown in FIG. 25.

[0044] FIG. 25 is a diagram illustrating the configura-

tion of a communication packet utilized in a communication interface of the printing system.

[0045] In FIG. 25, the communication packet is configured by a header portion H comprising 6 bytes, and data Data provided thereafter.

[0046] The header portion H will now be described in more detail.

[0047] The first byte represents Primary Socket ID (hereinafter abbreviated as "PSID"), which represents Socket ID of an apparatus where initialization of control according to the IEEE 1284.4 (to be described later) is started. The Socket ID will be described later.

[0048] The second byte represents Secondary Socket ID (hereinafter abbreviated as "SSID"), which represents Socket ID of another apparatus with respect to the above-described apparatus.

[0049] The third and fourth bytes represent the size of the entire packet inclusive of the 6-byte header. For example, in the case of a packet including 10-byte data, a value 16 [6 bytes for the header and 10 bytes for the data] is recorded in this portion. FIGS. 26A and 26B illustrate data which is provided in the form of a packet, and data which is not provided in the form of a packet, respectively.

[0050] FIGS. 26A and 26B are diagrams, each illustrating communication data exchanged through a communication interface of the printing system: FIG. 26A illustrates data which is provided in the form of a packet; and FIG. 26B illustrates data which is not provided in the form of a packet.

[0051] In the case of FIG. 25, for a packet having 0-byte data, a value 6 which represents 6 bytes for only the header portion is recorded in the portion of the third and fourth bytes.

[0052] Actually, however, since each data is represented by hexadecimal notation, the above-described 16 and 6 are represented by 0x0010 and 0x0006, respectively (in the case of hexadecimal notation, the notation generally starts with "0x", and 1 byte is represented by two digits).

[0053] The fifth byte represents Credit (to be described later). The sixth byte represents control data described as Control. One bit in this byte (one byte comprises eight bits) represents whether or not the packet is an ordinary packet (which is represented by "0"). Since the packet is an ordinary packet in most cases, the bit is 0. The remaining seven bits are used as a reserved region which is not used at present (represented by 0).

[0054] Accordingly, this byte is usually represented by "0x00". The data portion includes transfer data formed by application of each of the host apparatus and the printing apparatus (printing data and control data), or a control command in the IEEE 1284.4 (to be described later).

[0055] Next, the Socket ID shown in FIG. 25 will be described.

[0056] When a plurality of processing (mainly the processing of the application portion) in two appara-

tuses connected by a single interface (for example, an IEEE 1284 interface) identify respective partners thereof (processing in another apparatus) and exchange data independently, an individual number provided for each processing within each apparatus for executing the data exchange is represented by Socket ID. An outline of the Socket ID will now be described.

[0057] For example, when two types of processing (processing A and processing B) are present in the application of the host apparatus, and two types of processing (processing "a" for the processing A and processing "b" for the processing B) are present in the application of the printing apparatus, it is assumed that the PSIDs for the processing A and the processing B are 0x10 and 0x20, and the SSIDs for the processing "a" and the processing "b" are 0x10 and 0x20, respectively. By making the PSID and the SSID of the packet 0x10 and 0x10, respectively, when the processing A transmits data

"a"

and

"a",

and making the PSID and the SSID of the packet 0x20 and 0x20, respectively, when the processing B transmits data

"b"

and

"b",

according to the control of the IEEE 1284.4, it is possible to recognize from which processing to which processing the transmitted packet is transferred, and therefore to assuredly transfer the data to corresponding processing. Hence, it can be said that a plurality of channels have been logically used so as to realize a multichannel operation.

[0058] FIGS. 27 - 31 are diagrams illustrating the concept of a multichannel operation in a communication interface in the printing system, and sequentially illustrate a manner in which data are independently transferred from a plurality of (two in this case) application of the host apparatus to a plurality (two in this case) of corresponding application in the printing apparatus via respective processing.

[0059] Particularly, in FIG. 27, two types of processing (the processing A and the processing B) are present in the application of the host apparatus, and each

processing has data to be transferred

("a" and "b")

for the processing A, and

"a" and "b"

for the processing B).

[0060] FIG. 28 illustrates a manner in which packets are formed in the IEEE 1284.4 processing in the host apparatus, and the above-described Socket IDs are provided for each of the packets. The procedure for transferring the packets will be described later.

[0061] FIG. 29 illustrates a manner in which each data in the form of a packet is transferred from the host apparatus to the printing apparatus via the processing of the interface (IEEE 1284 processing). As shown in FIG. 29, the order of transfer of the packets is not particularly defined. In the case of FIG. 29, four packets are transferred in the order of the first packet of the processing A (the packet having the data

"a"),

the first packet of the processing B (the packet having the data

"b"),

the second packet of the processing A (the packet having the data

"a"),

and the second packet of the processing B (the packet having the data

"b").

[0062] FIG. 30 illustrates a manner in which the packets transferred to the printing apparatus are received by the IEEE 1284.4 at the printing apparatus side, and data extracted from each packet is transferred to corresponding application in accordance with the Socket IDs. As a result, as shown in FIG. 31, each data is transferred to the processing of corresponding application (the processing "a" and the processing "b") within the printing apparatus. A similar operation is, of course, performed for transfer in a direction reverse to the above-described case (from the processing in the printing

apparatus to the processing in the host apparatus).

[0063] When Socket IDs are represented by 0x00 (i.e., both PSID and SSID are represented by 0x00), only Command and Reply Packet (to be described later) necessary for executing the IEEE 1284.4 processing are transferred.

[0064] Next, the procedure (the flow of processing) of basic control in the IEEE 1284.4 will be described with reference to the flowcharts shown in FIGS. 32 - 34, and examples of packets shown in FIGS. 35 and 36.

[0065] FIGS. 32 - 34 are flowcharts illustrating a data communication processing procedure in the printing system. In FIGS. 32 - 34, numerals (1) - (24) represent respective steps.

[0066] First, an apparatus intending to start control according to the IEEE 1284.4 provides another apparatus with an initial command packet Init Command Packet (step (1)) (since a host apparatus usually provides a printing apparatus with such a packet, the apparatus which has provided the packet Init Command Packet is called the host apparatus, and another apparatus is called the printing apparatus).

[0067] The structure of the packet Init Command Packet will now be described with reference to FIG. 35.

[0068] FIG. 35 is a diagram illustrating the data structure of an initial command packet notified from the upper apparatus to the apparatus of the communication partner (the printing apparatus) in the printing system. The packet Init Command Packet includes the first through eighth bytes, and a header portion includes the first through sixth bytes.

[0069] In FIG. 35, data in portions other than a portion Credit in the fifth byte are substantially fixed data.

[0070] The printing apparatus which has received the packet Init Command Packet determines, if control according to the IEEE 1284.4 can be started (step (2)). If the result of the determination in step (2) is affirmative, a reply packet Init Reply Packet indicating the possibility of starting the control is formed (step (3)). If the result of the determination in step (2) is negative, a reply packet Init Reply Packet indicating the incapability of starting the control is formed (step (4)), and the process proceeds to step (5). At that time, if control according to the IEEE 1284.4 can be started, initialization for the start is also executed.

[0071] In step (5), the initial reply packet Init Reply Packet is transferred to the host apparatus. The data structure of the packet Init Reply Packet will now be briefly described with reference to FIG. 36.

[0072] FIG. 36 is a diagram illustrating the data structure of a reply packet notified from the printing apparatus to the host apparatus requesting communication in the printing system. The packet Init Reply Packet includes the first through ninth bytes, and a header portion includes the first through sixth bytes.

[0073] In FIG. 36, the eighth byte represents data Result. The reply packet Init Reply Packet notifies the host apparatus whether processing according to the

IEEE 1284.4 can be normally started or the processing cannot be started because of some reason.

[0074] Then, the host apparatus determines if the processing according to the IEEE 1284.4 can be started, i.e., if data processing is to be normally continued, in accordance with the notified packet Init Reply Packet (step (6)). If the result of the determination in step (6) is affirmative, the host apparatus then provides the printer with a command packet Open Channel Command Packet having a data structure as shown in FIG. 37.

[0075] If the result of the determination in step (6) is negative, the packet Init Command Packet is again transferred to the printing apparatus, or control according to the IEEE 1284.4 is given up.

[0076] The data structure of the packet Open Channel Command Packet will now be briefly described with reference to FIG. 37.

[0077] FIG. 37 is a diagram illustrating the data structure of a packet Open Channel Command Packet notified from the host apparatus to the printing apparatus in the printing system. The packet Open Channel Command Packet includes the first through seventeenth bytes, and a header portion includes the first through sixth bytes.

[0078] In FIG. 37, the ninth byte represents a parameter, which corresponds to Secondary Socket ID. The tenth and eleventh bytes also represent a parameter Primary To Secondary Packet Size indicating the maximum size of a packet transferred from the host apparatus to the printing apparatus according to a request of the host apparatus.

[0079] The twelfth and thirteenth bytes represent a parameter Secondary To Primary Packet Size indicating the maximum size of a packet transferred from the printing apparatus to the host apparatus according to a request of the host apparatus.

[0080] The fourteenth and fifteenth bytes represent a parameter which corresponds to information Credit Requested requested by the host apparatus. The sixteenth and seventeenth bytes represent a parameter which corresponds to information Maximum Outstanding Credit.

[0081] When the printing apparatus has received the packet Open Channel Command Packet, the printing apparatus confirms if the value of the parameter Secondary Socket ID in the received packet equals the value of the SSID possessed by the printing apparatus (step (8)). If the received SSID is confirmed to be the SSID possessed by the printing apparatus, the channel is established (opened).

[0082] It is possible to identify the opened channel based on the combination of Socket IDs (PSID and SSID). That is, as described above with respect to the Socket IDs, the channel connecting the processing A of the application in the host apparatus to the processing "a" of the application in the printing apparatus can be identified based on the combination of the Socket IDs

(PSID = 0x10 and SSID = 0x10). Similarly, the channel connecting the processing B of the application in the host apparatus to the processing "b" of the application in the printing apparatus can be identified based on the combination of the Socket IDs (PSID = 0x20 and SSID = 0x20).

[0083] Then, the parameter Primary To Secondary Packet Size indicating the maximum size of the packet when transferring data from the host apparatus to the printing apparatus according to a request of the host apparatus, and the parameter Secondary To Primary Packet Size indicating the maximum size of the packet when transferring data from the printing apparatus to the host apparatus according to a request of the host apparatus are checked, in order to determine if the checked size is a packet size which can be prepared by the corresponding apparatus (step (9)). If the result of the determination in step (9) is affirmative, a parameter indicating the fact is set in a packet Open Channel Reply Packet (to be described later). If the result of the determination in step (9) is negative, the packet sizes (Primary To Secondary Packet Size and Secondary To Primary Packet Size) which can be prepared by the respective apparatuses are set as the parameters.

[0084] Then, a parameter Credit Requested and a parameter Maximum Outstanding Credit representing information Credit requested by the host apparatus are analyzed (step (10)), and a reply for a request represented by the combination of the two parameters is performed using a parameter Open Channel Reply as in the above-described case of the packet size.

[0085] When the analysis of respective parameters Open Channel Command has been completed in the above-described manner, the printing apparatus forms a packet Open Channel Reply Packet shown in FIG. 38 (step (11)) and transfers the formed packet to the host apparatus (step (12)).

[0086] FIG. 38 is a diagram illustrating the data structure of the packet Open Channel Reply Packet notified from the printing apparatus to the host apparatus in the printing system. The packet Open Channel Reply Packet includes the first through sixteenth bytes, and a header portion includes the first through sixth bytes.

[0087] In FIG. 38, the eleventh and twelfth bytes represent a parameter Primary To Secondary Packet Size indicating the maximum size of the packet transferred from the host apparatus to the printing apparatus according to a request of the host apparatus.

[0088] The thirteenth and fourteenth bytes represent a parameter Secondary To Primary Packet Size indicating the maximum size of the packet transferred from the printing apparatus to the host apparatus according to a request of the host apparatus.

[0089] Then, the host apparatus determines if the channel has been opened by analyzing the parameter Open Channel Reply Packet (step (13)). If the result of the determination in step (13) is affirmative, the data packet having the data structure shown in FIG. 25 can

be transferred to the opened channel (step (14)).

[0090] The size of the data packet to be transferred is determined by comparing the packet size requested by the user (the host apparatus which has provided the packet Open Channel Command Packet) with the packet size requested by the printing apparatus side and indicated by the reply Open Channel Reply, and adopting the smaller packet size.

[0091] The transfer of the data packet in the reverse direction (i.e., from the printing apparatus to the host apparatus) may, of course, be performed, and the transfer of the packet Command Packet relating to Credit (to be described later) becomes effective in this case (the transfer may, of course, be performed before opening the channel, though such transfer is meaningless).

[0092] Any number of channels may be opened provided that processing can be performed within the apparatus (and combination of Socket IDs does not overlap), by repeating the return to step (7). In the above-described example of Socket IDs, two channels, i.e., the channel connecting the processing A of the host apparatus to the processing "a" of the printing apparatus, and the channel connecting the processing B to the processing "b", are opened.

[0093] The flow until the channel connecting the processing of application within the host apparatus to the processing of application within the printing apparatus is opened has been described. Since a channel for transferring various commands Command and replies Reply (Init. Open Channel and the like) (hereinafter termed a "command channel") has been implicitly opened, the above-described procedure is unnecessary.

[0094] As has been described, the Socket ID for identifying the channel is "0x00". It is possible to terminate (close) a channel where data exchange has been terminated (which is to be terminated) by a packet Close Channel Command Packet.

[0095] When the host apparatus intends to close a channel, a packet Close Channel Command Packet having the data structure shown in FIG. 39 is transferred to the printing apparatus (step (15)).

[0096] FIG. 39 is a diagram illustrating the data structure of a packet Close Channel Command Packet notified from the host apparatus to the printing apparatus in the printing system. The packet Close Channel Command Packet includes the first through ninth bytes, and a header portion includes the first through sixth bytes.

[0097] In FIG. 39, the eighth and ninth bytes represent a parameter which is obtained by combining Primary Socket ID and Secondary Socket ID for indicating the channel to be closed.

[0098] The printing apparatus analyzes the parameter (step (16)), and determines if the channel assigned to be closed is closable (opened) (step (17)). If the result of the determination in step (16) is affirmative, the printing apparatus closes the channel, and prepares a reply for notifying the host apparatus of the fact using a

packet Close Channel Reply Packet (step (18)).

[0099] If the result of the determination in step (17) is negative, a packet Reply Packet is formed (step (19)).

[0100] After thus forming the packet Close Channel Reply Packet, the printing apparatus transfers the formed packet Close Channel Reply Packet having the data structure shown in FIG. 40 to the host apparatus (step (20)).

[0101] FIG. 40 is a diagram illustrating the data structure of the packet Close Channel Reply Packet notified from the printing apparatus to the host apparatus in the printing system. The packet Close Channel Reply Packet includes the first through tenth bytes, and a header portion includes the first through sixth bytes.

[0102] In FIG. 40, a parameter Result is set in the eighth byte. The ninth and tenth bytes represent a parameter including Primary Socket ID and Secondary Socket ID.

[0103] The host apparatus confirms that the channel is closed by receiving the packet Close Channel Reply Packet and analyzing the above-described parameter Result (step (21)).

[0104] After the channel has been closed, there is no problem to again open the channel by a packet Open Channel Command Packet if necessary (return to step (7)), or the control according to the IEEE 1284.4 may be terminated in that state.

[0105] In this case, the host apparatus provides a packet Exit Command Packet having the data structure shown in FIG. 41 and transfers this packet to the printing apparatus (step (22)).

[0106] FIG. 41 is a diagram illustrating the data structure of the packet Exit Command Packet notified from the host apparatus to the printing apparatus in the printing system. The packet Exit Command Packet includes the first through seventh bytes, and a header portion includes the first through sixth bytes.

[0107] In FIG. 41, a command Exit Command is set in the seventh byte.

[0108] When the printing apparatus has received the packet Exit Command Packet, the printing apparatus terminates the processing according to the IEEE 1284.4 (data exchange on the opened channel, and the like) (step (23)), and then forms a reply packet Exit Reply Packet having the data structure shown in FIG. 42 and transfers the formed packet to the host apparatus (step (24)).

[0109] FIG. 42 is a diagram illustrating the data structure of the packet Exit Reply Packet notified from the printing apparatus to the host apparatus in the printing system. The packet Exit Reply Packet includes the first through eighth bytes, and a header portion includes the first through sixth bytes.

[0110] In FIG. 42, a command Command is set in the seventh byte, and a result Result is set in the eighth byte.

[0111] The control according to the IEEE 1284.4 is thereby terminated, and the data transfer by the packet

(conforming to the IEEE 1284.4) is also terminated.

[0112] When resuming the control according to the IEEE 1284.4, it is necessary to execute the control from processing of transferring the packet Init Command Packet. The command packet Exit Command Packet may, of course, be provided at any time while the control according to the IEEE 1284.4 is executed, as any other command packet Command Packet.

[0113] By executing the process until step (14) of the above-described flowcharts in order to open the channel, the transfer of the data packet can be started. However, the control is performed so that, when transferring the data packet, the transfer cannot be started unless acknowledging data Credit is not received from the processing of the apparatus where the data packet is to be transferred.

[0114] Thus, the IEEE 1284.4 realizes flow control of data (packet) (as for the Socket ID, when the processing A at the host apparatus side transfers the packet to the processing "a" at the printing apparatus side, the processing A cannot process the packet unless a parameter Credit is not received from the processing "a"). When storing data transferred from the communication partner in resources (storage means, capable of reading and writing data, which is generally called a buffer storage or the like, and which is hereinafter termed a reception buffer storage assuming the printing apparatus) possessed by the user, the processing will not be normally operated by missing transferred data, unless control is performed so as not to receive data to an amount exceeding the capacity of the reception buffer storage. The flow control adjusts interruption and resumption of data transfer in order to prevent such a situation.

[0115] Next, a description will be provided of the concept of the parameter Credit used in the flow control of packet transfer in the IEEE 1284.4.

[0116] The parameter Credit is provided from the packet receiving side to the packet transmitting side in order to indicate to what extent the receiving side is ready to receive the packet. The packet receiving side guarantees that the packet provided as the parameter Credit can be assuredly received.

[0117] For the exchange of the parameter Credit, an acknowledgment command packet Credit Command Packet and an acknowledgment request command packet Credit Request Command Packet and a packet Reply Packet corresponding to these packets are mainly used. However, the parameter Credit may also be provided by a packet Data Packet or Open Packet, except for a command channel which already has two parameters Credit at initialization, in order to allow execution of Init or Error processing according to the IEEE 1284.4 in any case.

[0118] Next, state transition of a simple parameter Credit will be described with reference to FIG. 43.

[0119] FIG. 43 is a diagram illustrating state transition of the parameter Credit between the host apparatus and



the printing apparatus in the printing system.

[0120] In an initial state in which power is supplied to the apparatuses (the host apparatus and the printing apparatus), the parameter Credit on the internal command channel is set to "2", and the parameter Credit on the channel (which is not yet opened at this stage) for transferring data (a packet) is, of course, set to "0".

[0121] In this example, only one channel for transferring data is shown. When a plurality of channels are opened, the control of the parameter Credit is performed for each channel.

[0122] When the host apparatus intends to start control according to the IEEE 1284.4 using an initial command packet Init Command Packet, the parameter Credit on the command channel at the host apparatus side is decremented by one to assume "1" according to the provision of the packet Init Command Packet. At the same time, at the printing apparatus side, when the packet Init Command Packet has been received, the acknowledgment Credit on the command channel is incremented by one to assume "3". This is because it is promised that "1" is always present in the parameter Credit (the fifth byte of the header portion) of the packet transferred on the command channel. If the parameter Credit assumes "0" on the command channel and therefore the packet cannot be transferred, the succeeding processing cannot be performed. The above-described promise is provided in order to prevent such a case.

[0123] By transmitting a reply packet Init Reply Packet in response to the packet Init Command Packet from the printing apparatus, the parameter Credit on the command channel of the printing apparatus is decremented by one to assume "2", and the parameter Credit on the command channel of the host apparatus is incremented by one to return to "2".

[0124] When the host apparatus intends to open a data channel according to the packet Open Channel Command Packet, and the printing apparatus normally transmits a reply Open Channel Reply in response to this operation, by providing the host apparatus with two parameters Credit using the parameter Credit (the fifteenth and sixteenth bytes) in the packet Open Channel Reply Packet, the parameter Credit on the channel for transferring data at the host apparatus side is incremented by two to assume "2".

[0125] Since the host apparatus has been able to open the channel for transferring data and has received the parameter Credit for this channel from the printing apparatus, it becomes possible to transfer a packet Data Packet. Hence, the host apparatus transfers two consecutive packets Data Packet. For the second packet Data Packet, a parameter Credit on the channel for transferring data to the printing apparatus is provided using the parameter Credit (the fourth byte) of the packet Data Packet. At that time, the parameter Credit on the channel for transferring data in the printing apparatus is incremented by one to assume "1".

[0126] The parameter Credit on the command chan-

nel remains "2" because a packet Command Packet is not transferred (the printing apparatus does not transfer a packet Command Packet either). At that time, the printing apparatus can also transfer a packet Data Packet to the host apparatus.

[0127] The host apparatus has transferred two consecutive packets Data Packet and therefore used up parameters Credit on the channel for transferring data possessed by the host apparatus. Hence, in order to transfer the next packet Data Packet, the host apparatus requests a parameter Credit from the printing apparatus using a packet Credit Request Command Packet having the data structure shown in FIG. 44.

[0128] FIG. 44 is a diagram illustrating the data structure of the packet Credit Request Command Packet for requesting a parameter Credit in the printing system. This packet includes the first through thirteenth bytes, and a header portion includes the first through sixth bytes.

[0129] In FIG. 44, a parameter request Credit Requested is set in the tenth and eleventh bytes.

[0130] The printing apparatus to which a parameter Credit has been requested by the packet Credit Request Command Packet analyzes the contents of the request (from the parameter), and transfers a reply Reply corresponding to the request. In this case, it is assumed that only one parameter Credit can be provided for the host apparatus. The provided parameter Credit is entered in the parameter Credit (the eleventh and twelfth bytes) of the reply Credit Request Reply, and is transferred to the host apparatus by a packet Credit Request Reply Packet having the data structure shown in FIG. 45.

[0131] FIG. 45 is a diagram illustrating the data structure of the packet Credit Request Reply Packet in the printing system. This packet includes the first through twelfth bytes, and a header portion includes the first through sixth bytes.

[0132] In FIG. 44, the parameter Credit is set in the eleventh and twelfth bytes.

[0133] By thus receiving the packet Credit Request Reply Packet, the parameter Credit on the channel for data transfer of the host apparatus is incremented by one to assume "1", so that it is possible to again transfer a packet Data Packet.

[0134] Since a parameter Credit for data transfer has been provided from the host apparatus, the printing apparatus can also transfer one packet Data Packet.

[0135] As shown in FIG. 43, if the printing apparatus transfers one packet Data Packet, the parameter Credit possessed by the printing apparatus is decremented by one to assume "0". It is, of course, possible to provide the host apparatus with a parameter Credit using the parameter Credit in this packet Data Packet. The number of parameters Credit possessed by the host apparatus increases by the provided parameter Credit.

[0136] In FIG. 43, the parameter Credit for data transfer is incremented by one to assume "2".

[0137] An outline of the IEEE 1284.4 has been described.

[0138] However, since conventional nonpacket protocols are also used in addition to packet protocols in the current Centronics I/Fs, a packet protocol is used in certain data transfer, and data is transferred according to a conventional nonpacket protocol without using a packet protocol in another data transfer.

[0139] In such a case, if data transferred using a packet protocol is interpreted with a nonpacket protocol, a header portion is considered to be data, so that unnecessary extra data is received.

[0140] On the other hand, if data transferred using a non-packet protocol is interpreted with a packet protocol, an ordinary data portion is considered to be a packet header, so that a part of data of the packet is not given to the translator, thereby causing various problems.

[0141] A single printer is often shared by a plurality of host apparatuses. In such a case, a certain host apparatus may transmit data using a packet protocol, and another host apparatus may transfer data using a conventional nonpacket protocol, so that the printer cannot determine whether or not a packet protocol is to be used.

[0142] From the structure of a packet protocol, it is difficult to shift to a nonpacket protocol when the packet protocol has been used, and vice versa.

#### SUMMARY OF THE INVENTION

[0143] The present invention has been made in consideration of the above-described problems.

[0144] It is a concern of the present invention to provide a communication control apparatus, a communication control method and a storage medium storing a communication control program in which, when packet data has been received from a data processing apparatus, the packet data is stored in a first area or a second area of a memory in accordance with a type of data in order to use a packet protocol, and when nonpacket data has been received from the data processing apparatus, the nonpacket data is stored in the first area of the memory in order to use a nonpacket protocol.

[0145] According to one aspect, the present invention relates to a communication control apparatus for performing two-way communication processing with a data processing apparatus via a predetermined interface. The communication control apparatus includes storage means having a first area and a second area for storing data received from the data processing apparatus, and control means for storing, when packet data has been received from the data processing apparatus, the packet data in the first area or the second area in accordance with a type of data in order to use a packet protocol, and for storing, when nonpacket data has been received from the data processing apparatus, the nonpacket data in the first area in order to use a non-

packet protocol.

[0146] According to another aspect, the present invention relates to a communication control method in a communication control apparatus for performing two-way communication processing with a data processing apparatus via a predetermined interface. The communication control method includes the steps of storing, when packet data has been received from the data processing apparatus, the packet data in a first area or a second area of a memory in accordance with a type of data in order to use a packet protocol, and storing, when nonpacket data has been received from the data processing apparatus, the nonpacket data in the first area in order to use a nonpacket protocol.

[0147] According to still another aspect, the present invention relates to a storage medium storing a control program to be used in a communication control apparatus for performing two-way communication processing with a data processing apparatus via a predetermined interface. The storage medium includes a code of storing, when packet data has been received from the data processing apparatus, the packet data in a first area or a second area of a memory in accordance with a type of data in order to use a packet protocol, and a code of storing, when nonpacket data has been received from the data processing apparatus, the nonpacket data in the first area in order to use a nonpacket protocol.

[0148] The foregoing and other concerns, advantages and features of the present invention will become more apparent from the following description of the preferred embodiments taken in conjunction with the accompanying drawings.

#### BRIEF DESCRIPTION OF THE DRAWINGS

[0149]

FIG. 1 is a block diagram illustrating the configuration of a printing apparatus according to a first embodiment of the present invention;

FIG. 2 is a plan view illustrating an example of the panel device shown in FIG. 1;

FIG. 3 is a diagram illustrating a procedure for setting a packet protocol in a communication control apparatus according to the first embodiment;

FIGS. 4 through 11 are diagrams illustrating operational picture surfaces showing a procedure for setting a packet protocol in the communication apparatus according to the first embodiment;

FIG. 12 is a flowchart illustrating a data processing procedure in the communication control apparatus according to the first embodiment;

FIG. 13 is a block diagram illustrating the configuration of a printing system to which a communication control apparatus can be applied according to a second embodiment of the present invention;

FIG. 14 is a flowchart illustrating a data processing procedure in the communication control apparatus

according to the second embodiment;

FIG. 15 is a block diagram illustrating the configuration of a printing system to which a printing control apparatus can be applied according to a third embodiment of the present invention;

FIGS. 16 and 17 are timing charts illustrating a state of data communication in a communication control apparatus according to the third embodiment;

FIG. 18 is a flowchart illustrating a data processing procedure in the communication control apparatus according to the third embodiment;

FIGS. 19 and 20 are diagrams, each illustrating processing of moving received data in the communication control apparatus according to the third embodiment;

FIG. 21 is a diagram illustrating a memory map of a storage medium storing various data processing programs capable of being read by the communication control apparatus according to the third embodiment;

FIG. 22 is a plan view illustrating the configuration of a pin arrangement of a printer interface in a printing system to which a communication control apparatus can be applied;

FIG. 23 is a timing chart illustrating a state of data transfer processing in the printing system;

FIG. 24 is a diagram illustrating the concept of the relationship between each communication control program and application in the printing system;

FIG. 25 is a diagram illustrating the configuration of a communication packet utilized in a communication interface of the printing system;

FIGS. 26A and 26B are diagrams illustrating examples of communication data exchanged in the communication interface of the printing system;

FIGS. 27 through 31 are diagrams illustrating the concept of a multichannel operation in the communication interface of the printing apparatus;

FIGS. 32 through 34 are flowcharts illustrating a data communication processing procedure in the printing system;

FIG. 35 is a diagram illustrating the data structure of an initial command packet notified from a host apparatus to an apparatus serving as a communication partner (a printing apparatus) in the printing system;

FIG. 36 is a diagram illustrating the data structure of a reply packet notified from the printing apparatus to the host apparatus in the printing system;

FIG. 37 is a diagram illustrating the data structure of an open command packet notified from the host apparatus to the printing apparatus in the printing system;

FIG. 38 is a diagram illustrating the data structure of an open channel reply packet notified from the printing apparatus to the host apparatus in the printing system;

FIG. 39 is a diagram illustrating the data structure of a close channel command packet notified from the host apparatus to the printing apparatus in the printing system;

FIG. 40 is a diagram illustrating the data structure of a close channel reply packet notified from the printing apparatus to the host apparatus in the printing system;

FIG. 41 is a diagram illustrating the data structure of an exit command packet notified from the host apparatus to the printing apparatus in the printing system;

FIG. 42 is a diagram illustrating the data structure of an exit reply packet notified from the printing apparatus to the host apparatus in the printing system;

FIG. 43 is a diagram illustrating state transition of a parameter Credit between the host apparatus and the printing apparatus in the printing system;

FIG. 44 is a diagram illustrating the data structure of a parameter request command packet for requesting the parameter Credit in the printing system; and

FIG. 45 is a diagram illustrating the data structure of a parameter request reply packet in the printing system.

#### DESCRIPTION OF THE PREFERRED EMBODIMENTS

##### First Embodiment

[0150] FIG. 1 is a block diagram illustrating the configuration of a printing system to which a printing apparatus can be applied according to a first embodiment of the present invention. For example, a laser-beam printer is used as the printing apparatus.

[0151] In FIG. 1, a laser-beam printer (hereinafter abbreviated as an "LBP") 1 includes a printer controller 3 (hereinafter abbreviated as a "controller"), an engine 4, a panel device 5, and a disk device 6. The LBP 1 is connected to an external host apparatus 2, so as to execute printing processing by receiving printing data and the like from the host apparatus 2.

[0152] In the controller 3, a CPU (central processing unit) 7 controls respective units based on various control programs stored in a PROM (programmable read-only memory). The PROM 8 also stores programs to be referred to by the CPU 7 when determining a mode to be used during communication. The PROM 8 also stores programs to be referred to by the CPU 7 when determining a packet protocol to be used.

[0153] A host-apparatus interface (hereinafter an interface is abbreviated as an "I/F") 11 communicates printing data and the like with the host apparatus 2. An engine I/F 12 communicates with the engine 4 which actually performs printing. Reference numeral 13 represents a panel-device I/F. For example, the panel-device I/F 13 indicates the state of the LBP 1 to the user, or the user assigns a change in printing environment to the

LBP 1. A disk-device I/F 14 communicates with the disk device 6.

[0154] Reference numeral 9 represents an insertable/drawable optional memory for storing fonts and the like. A card, an optional ROM, a flash memory or the like may be adopted as the optional memory 9. Reference numeral 10 represents a RAM (random access memory). Regions, such as a frame buffer storage 10a for temporarily storing image data, a reception buffer storage 10b for temporarily storing data input from the host apparatus 2, such as printing data and the like, a printing-data buffer storage 10c for storing printing data, a control-command buffer storage 10d for storing control commands, and the like, are secured in the RAM 10. The RAM 10 is also used as a working area for the CPU 7.

[0155] Reference numeral 15 represents an NV(non-volatile)RAM. Regions, such as a packet protocol flag 15a for storing setting whether or not a packet protocol is to be used, and the like, are secured in the NVRAM 15. The packet protocol flag 15a has a value "0" or "1". The user can set the value "0" or "1" through an operation on the panel device 5 (to be described later). The NVRAM 15 is also used as a region for storing various setting values.

[0156] The panel device 15 includes, for example, a liquid-crystal panel display for displaying information relating to the state of the LBP 1 or the like in the form of a character string, various operation buttons for allowing the operator to perform various operations for the LBP 1, and LEDs (light-emitting diodes) for notifying the operator of a sheet feeding location, a on-line/off-line state and the like. The operator can also set whether or not a packet protocol is to be used via the panel device 5.

[0157] The disk device 6 is an external storage device for storing various data. A hard-disk device, a magnetooptical-disk device, a floppy-disk device or the like may be used as the disk device 6. The engine 4 actually prints image data on a printing medium. Electric power is supplied to the LBP 1 from a power-supply unit (not shown).

[0158] FIG. 2 is a plan view illustrating an example of the panel device 5 shown in FIG. 1.

[0159] In FIG. 2, an on-line LED 7-1 indicates whether the LBP 1 is in an on-line state or in an off-line state. An on-line key 7-2 is used for causing shift between the on-line state and the off-line state. A middle-stage sheet feeding LED 7-3 is lit when middle-stage sheet feeding has been selected. A tray sheet feeding LED 7-4 is lit when tray sheet feeding has been selected.

[0160] An upper-stage sheet feeding LED 7-5 is lit when upper-stage sheet feeding has been selected. A printable LED 7-6 is lit when an error and the like are not generated, and printing can be performed. A job LED 7-7 is lit during a job.

[0161] An upward-arrow key (also operating as a mode key) 7-8 is used for movement in the upward

direction when operating a printing-environment menu. A rightward-arrow key (also operating as a menu key) 7-9 is used for movement in the rightward direction when operating the printing environment menu. In the ordinary off-line state, the key 7-9 is depressed when entering the printing environment menu.

[0162] A downward-arrow key (also operating as a utility key) is used for movement in the downward direction or for determining a menu value when operating the printing environment menu. A leftward-arrow key 7-11 (also operating as a utility key) is used for movement in the leftward direction when operating the printing environment menu.

[0163] An alarm LED 7-12 is lit when error is generated. A status display unit 7-13 displays, for example, a character string indicating the state of the printer, or the contents of the printing environment menu. A sheet-feeding selection key 7-15 is used for changing the sheet feeding location. A duplex-mode selection key 7-16 is used for setting duplex-mode printing. A duplex-mode LED 7-17 indicates whether or not the printing apparatus is currently set to the duplex-mode printing. A discharge key 7-18 is used when it is intended to forcibly discharge printed data.

[0164] FIG. 3 illustrates a tree indicating an outline of the printing environment menu for setting whether or not the packet protocol is to be used.

[0165] FIG. 3 illustrates a procedure for setting the packet protocol in the communication control apparatus in the form of a tree. In this tree, setup is a "parent", followed by "children", "grandchildren", . . . in the downward direction.

[0166] It can be understood from FIG. 3 that, in order to operate setting values for the "packet protocol", after entering the printing environment menu in the panel device 5 shown in FIG. 1, "interface", "Centronix setting" and "packet protocol" may be set.

[0167] When "to use" has been set for the "packet protocol", a value "1" is set in the packet protocol flag 15a shown in FIG. 1. On the other hand, when "not to use" has been set for the "packet protocol", a value "0" is set in the packet protocol flag 15a secured in the NVRAM 15 shown in FIG. 1 in a nonvolatile state.

[0168] The characteristic configuration of the first embodiment will now be described with reference to FIG. 1 and other figures.

[0169] A communication control apparatus (the controller 3) for performing two-way communication processing with a data processing apparatus (the host apparatus 2) via a predetermined interface having the above-described configuration (in the first embodiment, a two-way communication interface conforming to the IEEE 1284.4, such as a Centronics interface) includes a storage unit capable of securing a plurality of memory regions for storing data received from the data processing apparatus (the RAM 10 where the frame buffer storage 10a, the reception buffer storage 10b, the printing-data buffer storage 10c, the control-command buffer

storage 10d and the like can be secured), a setting unit for setting the type of a protocol to be used for the communication processing (in the first embodiment, the setting is performed through the panel device 5, although the setting may be performed by a command from the host apparatus 2), a determination unit for determining whether or not the type of the protocol set by the setting unit indicates a packet protocol (the CPU 7 determines from the state of setting of the packet-protocol flag 15a by executing a control program stored in the PROM 8), and a control unit for changing destination of each data by determining the type of data in the received data stored in the storage unit when the determination unit has determined that the type of the set protocol indicates a packet protocol (the CPU 7 controls the destination of the data to one of the printing-data buffer storage 10c and the control-command buffer storage 10d by executing a control program stored in the PROM 8). Hence, the user or the like can set whether or not a specific packet protocol is to be used, and it is possible to freely construct a data communication environment capable of flexibly dealing with the user's request of packet communication.

[0170] The CPU 7 discriminates control commands from data other than the control commands in a portion other than the header portion within the received data, and moves the respective data to different memory regions (the control-command buffer storage 10d and the printing-data buffer storage 10c). Hence, it is possible to discriminate control commands peculiar to the packet protocol from other data, and normally process the respective data. The control commands peculiar to the packet protocol indicate job control commands for controlling respective jobs transmitted by packet communication, such as a job cancelling command and the like.

[0171] The communication control apparatus also includes a nonvolatile memory (the NVRAM 15) for storing, for example, the type of the protocol set by the panel device 5 in a non-volatile manner. Hence, it is possible to assuredly start data communication in which the communication environment set by the user is faithfully reflected.

[0172] Since the data other than the control commands is used as printing data, it is possible to assuredly prevent a case in which a control command is processed as printing data, and to perform processing of printing data for obtaining a normal printing result.

[0173] A description will now be provided of a setting operation and the contents of display on the status display unit 7-13 on the panel device 5 when using the packet protocol, with reference to FIGS. 4 - 11.

[0174] FIGS. 4 - 11 are diagrams illustrating operational picture surfaces indicating a procedure for setting a packet protocol in the communication control apparatus according to the first embodiment.

[0175] The status display unit 7-13 in the ordinary off-line state is as shown in FIG. 4. By depressing the right-

ward-arrow key 7-9 shown in FIG. 2, a printing environment menu is provided, and the display "00 printable A4" shown in FIG. 4 appears on the status display unit 7-13.

[0176] The first line L1 and the second line L2 on the status display unit 7-13 is in the relationship of a parent and a child. That is, in this case, a parent "setup" and a child "expansion function" are displayed. Referring to FIG. 3, the target "packet protocol" is lower than "interface", and the "interface" is present at the right of the "expansion function". Hence, the menu is moved to the right by depressing the rightward-arrow key 7-9. Then, a display of "setup" and "expansion function" shown in FIG. 5 is provided on the status display unit 7-13.

[0177] In order to display items below the "interface" shown in FIG. 3 from this display state, the downward-arrow key 7-10 is depressed, and the status display unit 7-13 provides the display picture surface shown in FIG. 6. By further depressing the downward-arrow key 7-10, the status display unit 7-13 provides the display picture surface shown in FIG. 7. By further depressing the downward-arrow key 7-10, the status display unit 7-13 provides the display picture surface shown in FIG. 8. By further depressing the downward-arrow key 7-10, the status display unit 7-13 provides the display picture surface shown in FIG. 9.

[0178] The symbol "=" displayed on the picture surface shown in FIG. 9 indicates that the "packet protocol" is set to "not to use".

[0179] By depressing the rightward-arrow key 7-9 in this state, the status display unit 7-13 provides the picture surface shown in FIG. 10, where "to use" can be selected. By depressing the downward-arrow key 7-10 for performing determination, the status display unit 7-13 provides the display picture surface shown in FIG. 11, where it is indicated that the "packet protocol" is set to "to use".

[0180] At that time, a value "1" is set in the packet-protocol flag 15a in the NVRAM 15 shown in FIG. 1. Similarly, by setting "not to use" for the "packet protocol", a value "0" is set in the packet-protocol flag 15a.

[0181] An operation from the transmission of data from the host apparatus to the printer to the execution of printing processing will now be described. In order to simplify the description, it is assumed that in a conventional nonpacket mode, only printing data is transmitted, and control commands in the first embodiment are not transmitted. Control commands (an image drawing command, a sheet discharging command and the like) included in the printing data are different from control commands in the first embodiment.

[0182] When data is input from the host apparatus 2 to the LBP 1 via the host-apparatus I/F 11, notification of interrupt instructing storage of input data is performed for the CPU 7. Processing for storing input data which is executed by the controller 3 in this case will now be described with reference to the flowchart shown in FIG. 12.

[0183] FIG. 12 is a flowchart illustrating a data processing procedure in the communication apparatus according to the first embodiment. In FIG. 12, numerals (1) - (4) represent respective steps.

[0184] When notification of interrupt for instructing storage of input data has been provided from the host apparatus 2, then, in step (1), the CPU 7 stores the input data into the reception buffer storage 10b via the host-apparatus I/F 11. When all the input data has been stored, then, in step (2), the value of the packet-protocol flag 15a in the NVRAM 15 shown in FIG. 1 is referred to. Different control is performed depending on whether the value of the packet-protocol flag 15a is "0" or "1". That is, when the packet-protocol flag 15a has a value "0", that indicates that the packet protocol is not to be used. Hence, the process proceeds to step (3), where data in the reception buffer storage 10b is moved to the printing-data buffer storage 10c. Then, the interrupt processing is terminated.

[0185] On the other hand, when the value of the packet-protocol flag 15a is "1", that indicates that the packet protocol is to be used. Hence, the process proceeds to step (4), where the data of the reception buffer storage 10b is moved to the printing-data buffer storage 10c or the control-command buffer storage 10d in accordance with information indicated in the packet header. At that time, only a portion other than the packet header is moved. Then, the interrupt processing is terminated.

[0186] As described above, according to interrupt occurred every time data is input from the host apparatus 2, the input data is stored in the reception buffer storage 10b. Then, in accordance with the value of the packet-protocol flag 10a or information in the packet header, the input data is stored in the printing-data buffer storage 10c or the control-command buffer storage 10d.

[0187] Next, a description will be provided of an outline of printing processing performed by the LVP 1 when input data from the host apparatus 2 is stored in the printing-data buffer storage 10c according to interrupt for storing the input data, with reference to FIG. 1.

[0188] When the CPU 7 has recognized that the input data has been stored in the printing-data buffer storage 10c, the CPU 7 interprets printing data stored in the printing-data buffer storage 10c by referring to a printing-data analysis program stored in the PROM 8, and converts the printing data into image data. The image data provided by the CPU 7 is stored into the frame buffer storage 10a.

[0189] The CPU 7 continues the processing by referring to the printing-data analysis program in the above-described manner. When the CPU 7 has found a sheet discharging command in the printing data, the CPU 7 transmits the image data stored in the frame buffer storage 10a to the engine 4 via the engine I/F 12. Upon reception of the image data transmitted from the CPU 7 via the engine I/F 12, the engine 4 provides printing out-

put of the image data on an actual recording medium.

[0190] As described above, when data has been input from the host apparatus 2, the LBP 1 performs printing processing, and image data is actually output on a recording medium.

[0191] As described above, when a value "1" is stored in the packet-protocol flag 15a, the printing apparatus operates using the packet protocol, and when a value "0" is stored in the packet-protocol flag 15a, the printing apparatus operates using a conventional nonpacket protocol without using the packet protocol.

[0192] That is, by referring to the packet-protocol flag 15a preset by the user, and performing an operation in accordance with the contents of the packet-protocol flag 15a, it is possible to perform an operation as desired by the user.

[0193] Although in the first embodiment, the case of setting a mode to be used through the panel device 5 has been described, the setting means is not limited to such a device, and the mode may be set by any other appropriate means. For example, the mode may be set by a command instructed from the host apparatus 2.

[0194] A description will now be provided of an example in which the mode of using a packet protocol is set by an instruction from the host apparatus 2 in a situation in which the packet protocol is not used, with reference to FIG. 1.

[0195] When data "SET PACKET PROTOCOL = USE  $\forall n$ " representing a command for setting the use of a packet protocol has been transmitted from the host apparatus 2, notification of interrupt for instructing storage of input data is performed for the CPU 7, as described above.

[0196] The CPU 7 stores the input data "SET PACKET PROTOCOL = USE  $\forall n$ " in the reception buffer storage 10b via the host-apparatus I/F 11, and refers to the value of the packet-protocol flag 15a after completing the storage processing. Since setting of not using a packet protocol is provided in this case, the CPU 7 moves the data stored in the reception buffer storage 10b to the printing-data buffer storage 10c.

[0197] The CPU 7 temporarily terminates interrupt. Then, the CPU 7 recognizes that the input data has been stored in the printing-data buffer storage 10c, and analyzes the data "SET PACKET PROTOCOL = USE  $\forall n$ " stored in the printing-data buffer storage 10c by referring to the analysis program stored in the PROM 8. The data is interpreted as a command for setting the use of a protocol by the analysis program, and the CPU 7 sets a value "1" in the packet-protocol flag 15a.

[0198] Thus, it is possible to set a mode to be used according to a command instructed from the host apparatus 2.

[0199] It is also possible to provide setting of not using a packet protocol by an instruction from the host apparatus 2 when a packet protocol is used.

[0200] The characteristic configuration of the first embodiment will now be described with reference to

FIG. 12 and other figures.

[0201] A data processing method for a communication control apparatus for performing two-way communication processing with a data processing apparatus (the host apparatus 2) via a predetermined interface having the above-described configuration (in the first embodiment, a two-way communication interface conforming to the IEEE 1284.4, such as a Centronics interface), or a storage medium storing a program capable of being read by a computer for controlling an apparatus for performing two-way communication processing with a data processing apparatus via a predetermined interface includes a storage step of storing data received from the data processing apparatus in one of regions of a memory capable of securing a plurality of memory regions for storing the data (the reception buffer storage 10b in the RAM 10) (step (1) in FIG. 12), a setting step of setting the type of a protocol to be used for the communication processing (preprocessing of step (1) shown in FIG. 12, not shown), a determination step of determining whether or not the type of the protocol set in the setting step indicates a packet protocol (step (2) shown in FIG. 12), and a transfer step of transferring respective data to different regions in the memory (the printing-data buffer storage 10c and the control-command buffer storage 10d) by determining the type of data in the received data stored in some region of the memory when the determination step has determined that the type of the set protocol indicates a packet protocol (step (3) or step (4) shown in FIG. 12). Hence, the user or the like can set whether or not a specific packet protocol is to be used, and it is possible to freely construct a data communication environment capable of flexibly dealing with the user's request of packet communication.

#### Second Embodiment

[0202] In the first embodiment, the operation is performed in accordance with the packet-protocol flag 15a. However, in the case of a host-apparatus I/F having a plurality of communication modes, it is more useful if whether or not a packet protocol is to be used can be set for each communication mode. A second embodiment of the present invention which performs such an operation will now be described. In order to simplify the description, only data transmission from a host apparatus to a printer will be described.

[0203] A case in which a Centronics host-apparatus interface supports a compatible mode and an ECP mode as communication modes for transmitting data from a host apparatus to a printer.

[0204] FIG. 13 is a block diagram illustrating the configuration of a printing system to which a communication control apparatus can be applied according to the second embodiment. In FIG. 13, the same components as those shown in FIG. 1 are indicated by the same reference numerals.

[0205] In FIG. 13, a RAM 10 includes regions, such as

a frame buffer storage 10a for storing image data, a reception buffer storage 10b for temporarily storing data input from a host apparatus 2, such as printing data and the like, a printing-data buffer storage 10c for storing printing data, a control-command buffer storage 10d for storing control commands, and the like. In addition, a reception-mode flag 10e is secured in the RAM 10.

[0206] A value "0" or "1" is stored in the reception-mode flag 10e in accordance with the communication mode when storing input data in the reception-buffer storage 10b. The RAM 10 is also used as a working area for a CPU 7.

[0207] In the second embodiment, regions, such as a compatible-mode-packet-protocol flag 15b for storing setting whether or not a packet protocol is to be used in the compatible mode, an ECP-mode-packet-protocol flag 15c for storing setting whether or not a packet protocol is to be used in the ECP mode, and the like, are secured in an NVRAM 15.

[0208] Each of the compatible-mode-packet-protocol flag 15b and the ECP-mode-packet-protocol flag 15c has a value "0" or "1".

[0209] The user can set one of these values in each of the compatible-mode-packet-protocol flag 15b and the ECP-mode-packet-protocol flag 15c by an operation through a panel device 5. The NVRAM 15 is also used as a region for storing various setting values.

[0210] The characteristic configuration of the second embodiment will now be described with reference to FIG. 13 and other figures.

[0211] A communication control apparatus (a controller 3) for performing two-way communication processing with a data processing apparatus (the host apparatus 2) via a predetermined interface having the above-described configuration (in the second embodiment, a two-way communication interface conforming to the IEEE 1284.4, such as a Centronics interface) according to a plurality of communication modes (the compatible mode and the ECP mode) includes a storage unit capable of securing a plurality of memory regions for storing data received from the data processing apparatus (the RAM 10 where the frame buffer storage 10a, the reception buffer storage 10b, the printing-data buffer storage 10c, the control-command buffer storage 10d and the like are secured), a setting unit for setting the type of a protocol to be used for the communication processing (in the second embodiment, the setting is performed through the panel device 5, although the setting may be performed by a command from the host apparatus 2), a determination unit for determining whether or not the type of the protocol set by the setting unit for each communication mode indicates a packet protocol (the CPU 7 determines from the state of setting of the compatible-mode-packet-protocol flag 15b and the ECP-mode-packet-protocol flag 15c by executing a control program stored in a PROM 8), and a control unit for changing destination of each data by determining the type of data in the received data stored in the storage unit when the

determination unit has determined that the type of the protocol set for each communication mode indicates a packet protocol (the CPU performs control so as to transfer the data to one of the printing-data buffer storage 10c and the control-command buffer storage 10d by executing a control program stored in the PROM 8). Hence, the user or the like can set whether or not a specific packet protocol is to be used for each communication mode, and it is possible to freely construct a data communication environment capable of flexibly dealing with the user's request of packet communication.

[0212] The CPU 7 discriminates control commands from data other than the control commands in a portion other than the header portion within the received data, and moves the respective data to different memory regions (the control-command buffer storage 10d and the printing-data buffer storage 10c). Hence, it is possible to discriminate control commands peculiar to the packet protocol from other data, and normally process the respective data.

[0213] The communication control apparatus also includes a non-volatile memory (the compatible-mode-packet-protocol flag 15b and the ECP-mode-packet-protocol flag 15c in the NVRAM 15) for storing the type of the protocol set by the panel device 5 for each communication mode in a nonvolatile manner. Hence, it is possible to assuredly start data communication in which the communication environment set by the user is faithfully reflected.

[0214] Since the data other than the control commands is used as printing data, it is possible to assuredly prevent a case in which a control command is processed as printing data, and to perform processing of printing data for obtaining a normal printing result.

[0215] The communication modes include a first data communication mode (compatible mode) of communicating data from the host apparatus 2 to an apparatus having the communication control apparatus (for example, a printing apparatus), and a second communication mode of communicating data from the data processing apparatus to an apparatus having the communication control apparatus, and vice versa. Hence, it is possible to individually set whether or not a packet protocol is to be used in the first communication mode and in the second communication mode (ECP mode), and therefore to flexibly deal with a communication processing environment required by the user.

[0216] An operation from the transmission of data from the host apparatus to the printer to the execution of printing processing will now be described. In order to simplify the description, it is assumed that in a conventional nonpacket mode, only printing data is transmitted, and control commands are not transmitted.

[0217] When data is input from the host apparatus 2 to an LBP 1 via a host-apparatus I/F 11, notification of interrupt instructing storage of input data is performed for the CPU 7. In this notification of interrupt, a value "0" or "1" is stored in the reception-mode flag 10e.

[0218] That is, when data is received in the compatible mode a value "0" is stored in the reception-mode flag 10e, and when data is received in the ECP mode, a value "1" is stored in the reception-mode flag 10e. Processing of storing input data which is executed by the controller 3 will now be described with reference to the flowchart shown in FIG. 14.

[0219] FIG. 14 is a flowchart illustrating a data processing procedure in the communication apparatus according to the second embodiment. In FIG. 14, numerals (1) - (6) represent respective steps.

[0220] When notification of interrupt for instructing storage of input data has been provided from the host apparatus 2, then, in step (1), the CPU 7 stores the input data in the reception buffer storage 10b via the host-apparatus I/F 11. When all the input data has been stored, then, in step (2), the value of the reception-mode flag 10e is referred to. Different control is performed depending on whether the value of the reception-mode flag 15e is "0" or "1".

[0221] That is, when the reception-mode flag 15e has a value "0", that indicates that the data currently stored in the reception buffer 10b has been received in the compatible mode. Hence, the process proceeds to step (3), where the value of the compatible-mode-packet-protocol flag 15b is referred to. Different control is performed depending on whether the value of the compatible-mode-packet-protocol 15b is "0" or "1". That is, when it is determined that the compatible-mode-packet-protocol flag 15b has a value "0", that indicates that a packet protocol is not used. Hence, the process proceeds to step (4), where the data in the reception buffer storage 10b is moved to the printing-data buffer storage 10c. Then, the interrupt processing is terminated.

[0222] On the other hand, when it is determined in step (3) that the value of the compatible-mode-packet-protocol flag 15b is "1", that indicates that the packet protocol is to be used. Hence, the process proceeds to step (5), where the data of the reception buffer storage 10b is moved to the printing-data buffer storage 10c or the control-command buffer storage 10d in accordance with information indicated in the packet header. At that time, only a portion other than the packet header is moved. Then, the interrupt processing is terminated.

[0223] On the other hand, if it is determined in step (2) that the value of the reception-mode flag 10e is "1", that indicates that the data currently stored in the reception buffer storage 10b has been received in the ECP mode. Hence, the process proceeds to step (6), where the value of the ECP-mode-packet-protocol 15c is referred to.

[0224] Different control is performed depending on whether the value of the ECP-mode-packet-protocol flag 15c is "0" or "1". That is, when the value of the ECP-mode-packet-protocol 15c is "0", this indicates that a packet protocol is not to be used. Hence, the process proceeds to step (4), and the above-described processing is thereafter performed.



[0225] On the other hand, when it is determined in step (5) that the value of the ECP-mode-packet-protocol flag 15c is "1", that indicates that a packet protocol is to be used. Hence, the process proceeds to step (6), and the above-described processing is thereafter performed.

[0226] As described above, according to interrupt occurred every time data is input from the host apparatus 2, the input data is stored in the reception buffer storage 10b. Then, in accordance with the value of the reception-mode flag 10e, the value of the compatible-mode-packet-protocol flag 15b, the value of the ECP-mode-packet-protocol flag 15c, or information in the packet header, the input data is stored in the printing-data buffer storage 10c or the control-command buffer storage 10d.

[0227] As for printing processing performed by the LBP 1 when input data from the host apparatus 2 is stored in the printing-data buffer storage 10c according to interrupt for storing the input data, the same processing as that described in the first embodiment is performed, and image data is actually output onto a recording medium.

[0228] As described above, when receiving data from the host apparatus 2 in the compatible mode, i.e., when a value "1" is stored in the compatible-mode-packet-protocol flag 15b, the printing apparatus operates using a packet protocol, and when a value "0" is stored in the compatible-mode-packet-protocol flag 15b, the printing apparatus operates using a conventional nonpacket protocol without using a packet protocol.

[0229] When receiving data from the host apparatus 2 in the ECP mode, the printing apparatus operates using a packet protocol if a value "1" is stored in the ECP-mode-packet-protocol flag 15c, and operates using a nonpacket protocol without using a packet protocol if a value "0" is stored in the ECP-mode-packet-protocol 15c.

[0230] That is, in accordance with the used reception mode, the user refers to the compatible-mode-packet-protocol flag 15b or the ECP-mode-packet-protocol 15c which has been set in advance for each reception mode, and operates in accordance with the contents of the concerned flag. It is thereby possible to perform an operation desired by the user.

[0231] The second embodiment is particularly effective when whether or not a packet protocol is to be used depends on a communication mode, for example, in a case in which data transmission is performed using a conventional nonpacket protocol when the host apparatus side transmits data in the compatible mode, and data transmission is performed using a packet protocol when the host apparatus side transmits data in the ECP mode. This is because whether or not a packet protocol is to be used can be automatically determined if a packet flag for each communication mode is appropriately set.

[0232] Although in the second embodiment, the case

of setting the mode to be used from the panel device 5 has been described, the setting method is not limited to such a method, and setting may be performed by any other appropriate method. For example, setting may be performed by an instruction from the host apparatus (since an example of such setting has been described in the first embodiment, further description thereof will be omitted).

[0233] The characteristic configuration of the second embodiment will now be described with reference to FIG. 14 and other figures.

[0234] A data processing method for a communication control apparatus (the controller 3) for performing two-way communication processing with a data processing apparatus (the host apparatus 2) via a predetermined interface having the above-described configuration (in the second embodiment, a two-way communication interface conforming to the IEEE 1284.4, such as a Centronics interface) according to a plurality of communication modes, or a storage medium storing a program capable of being read by a computer for controlling an apparatus for performing two-way communication processing with a data processing apparatus via a predetermined interface according to a plurality of communication modes includes a storage step of storing data received from the data processing apparatus in one of regions of a memory capable of securing a plurality of memory regions for storing the received data (step (1) shown in FIG. 14), a setting step of setting the type of a protocol to be used for the communication processing for each communication mode (preprocessing of step (1) shown in FIG. 14, not shown), a determination step of determining whether or not the type of the protocol set in the setting step for each communication mode indicates a packet protocol (steps (2) and (3) shown in FIG. 14), and a transfer step of transferring respective data to different regions in the memory when the determination step has determined that the type of the protocol set for each communication mode indicates a packet protocol (step (4) or (6) shown in FIG. 14). Hence, the user or the like can set whether or not a specific packet protocol is to be used for each communication mode, and it is possible to freely construct a data communication environment capable of flexibly dealing with the user's request of packet communication.

### Third Embodiment

[0235] In the first and second embodiments, the case of operating in accordance with setting indicating whether or not a packet protocol is to be used has been described. However, it is also possible to determine whether or not a packet protocol is to be used according to whether a specific signal line assumes "High" or "Low". A third embodiment of the present invention which performs such an approach will now be described. In order to simplify the description, only data transmission from a host apparatus to a printer will be described.

[0236] FIG. 15 is a block diagram illustrating the configuration of a printing system to which a printing control apparatus can be applied according to the third embodiment. In FIG. 15, the same components as those shown in FIG. 1 are indicated by the same reference numerals. The configuration of FIG. 15 differs from that of FIG. 1 only in the RAM 10 and the NVRAM 15.

[0237] In FIG. 15, a RAM 10 includes regions, such as a frame buffer storage 10a for storing image data, a reception buffer storage 10b for temporarily storing data input from a host apparatus 2, such as printing data and the like, a printing-data buffer storage 10c for storing printing data, a control-command buffer storage 10d for storing control commands, and the like. In addition, a specific-signal-line flag 10f is secured in the RAM 10.

[0238] A value "0" or "1" is stored in the specific-signal-line flag 10f in accordance with whether or not the signal line when storing input data in the reception buffer storage 10b assumes "High" or "Low", respectively. The RAM 10 is also used as a working area for a CPU 7. An NVRAM 15 is used as a region for storing various setting values.

[0239] Various rules of selection can be considered as for which signal line is to be actually used as a "specific signal line" represented by the specific-signal-line flag 10f.

[0240] For example, as the first rule of selection in the case of a Centronics interface, a signal line which is in a state of "Not defined" according to the IEEE 1284 standards is used. This is because such a signal line is defined such that "the maker can use such a line on his own responsibility".

[0241] As the second rule of selection, a signal line which is not currently used and whose "High" or "Low" state does not influence any other lines, such as the line AUTO FD (a signal line corresponding to the fourteenth pin shown in FIG. 22), is used.

[0242] An operation from the transmission of data from the host apparatus 2 to an LBP 1 to the execution of printing processing will now be described. In order to simplify the description, it is assumed that in a conventional nonpacket mode, only printing data is transmitted, and control commands are not transmitted.

[0243] When data is input from the host apparatus 2 to the LBP 1 via a host-apparatus I/F 11, notification of interrupt instructing storage of input data is performed for the CPU 7. At this notification of interrupt, a value "1" or "0" is stored in the specific-signal-line flag 10f secured in the RAM 10 shown in FIG. 15, depending on whether the signal line assumes "High" or "Low", respectively.

[0244] That is, when the specific signal line assumes "Low", a value "0" is stored in the specific-signal-line flag 10f, and when the specific signal line assumes "High", a value "1" is stored in the specific-signal-line flag 10f.

[0245] A case in which the signal line AUTO FD is used as the specific signal line will now be described

with reference to FIGS. 16 and 17.

[0246] FIGS. 16 and 17 are timing charts, each illustrating a data communication state in a communication control apparatus according to the third embodiment. FIG. 16 corresponds to a case in which, when data is input from the host apparatus 2 to the LBP 1, a value "0" is stored in the specific-signal-line flag 10f secured in the RAM 10 because the signal line AUTO FD assumes "Low". FIG. 17 corresponds to a case in which, when data is input from the host apparatus 2 to the LBP 1, a value "1" is stored in the specific-signal-line flag 10f because the signal line AUTO FD assumes "High". The specific-signal-line flag 10f may be stored in the NVRAM 15.

[0247] In FIGS. 16 and 17, DATA1 - DATA8 are data lines, and correspond to information relating to the 0th - 7th bits, respectively, of data transmitted from the host apparatus 2.

[0248] DATA STROBE represents a data strobe line, which assumes a "High" state in a stationary state. When the DATA STROBE assumes a "Low" state, the LBP 1 reads the states of the data lines DATA1 - DATA8. ACK represents an acknowledge line, which assumes the "High" state in a stationary state. A pulse for providing the "Low" state is generated when the state shifts from the stationary state.

[0249] BUSY represents a busy line, on which a state signal indicating whether or not the LBP 1 can receive data from the host apparatus 2. The "Low" state represents a state in which a signal can be received, and the "High" state represents a state in which a signal cannot be received.

[0250] The characteristic configuration of the third embodiment will now be described with reference to FIG. 15 and other figures.

[0251] A communication control apparatus (a controller 3) for performing two-way communication processing with a data processing apparatus (the host apparatus 2) via a predetermined interface having a predetermined number of communication lines configured as described above (in the third embodiment, a two-way communication interface conforming to the IEEE 1284.4, such as a Centronics interface) includes a storage unit capable of securing a plurality of memory regions for storing data received from the data processing apparatus (the RAM 10 where the frame buffer storage 10a, the reception buffer storage 10b, the printing-data buffer storage 10c, the control-command buffer storage 10d and the like are secured), a detection unit for detecting the state of a specific communication line from among the plurality of communication lines (the CPU 7 detects "Low/High" of the specific signal line of the host-apparatus interface 11), a determination unit for determining whether or not the type of the protocol for data communication indicates a packet protocol (the CPU 7 determines from the state of setting of the specific-signal-line flag 10f set based on the result of the detection of "Low/High" of the specific signal line of the host-apparatus interface 11),

and a control unit for changing destination of each data by determining the type of data in the received data stored in the storage unit when the determination unit has determined that the type of the protocol indicates a packet protocol (the CPU 7 controls the destination of the data to one of the printing-data buffer storage 10c and the control-command buffer storage 10d by executing a control program stored in the PROM 8). Hence, it is possible to automatically determine and set whether or not a specific packet protocol is to be used, and to freely construct a data communication environment capable of flexibly dealing with a request of packet communication.

[0252] The CPU 7 discriminates control commands from data other than the control commands in a portion other than the header portion within the received data, and moves the respective data to different memory regions (the control-command buffer storage 10d and the printing-data buffer storage 10c). Hence, it is possible to discriminate control commands peculiar to the packet protocol from other data, and normally process the respective data.

[0253] Since the data other than the control commands is used as printing data, it is possible to assuredly prevent a case in which a control command is processed as printing data, and to perform processing of printing data for obtaining a normal printing result.

[0254] Processing for storing input data which is executed by the controller 3 will now be described with reference to the flowchart shown in FIG. 18.

[0255] FIG. 18 is a flowchart illustrating a data processing procedure in the communication control apparatus according to the third embodiment. In FIG. 18, numerals (1) - (4) represent respective steps.

[0256] When notification of interrupt for instructing storage of input data has been provided from the host apparatus 2, then, in step (1), the CPU 7 stores the input data into the reception buffer storage 10b via the host-apparatus I/F 11. When all the input data has been stored, then, in step (2), the value of the specific-signal-line flag 10f secured in the RAM 10 is referred to. Different control is performed depending on whether the value of the specific-signal-line flag 10f is "0" or "1". That is, when the specific-signal-line flag 10f is determined to have a value "0", that indicates that a packet protocol is not to be used. Hence, the process proceeds to step (3), where data in the reception buffer storage 10b is moved to the printing-data buffer storage 10c. Then, the interrupt processing is terminated.

[0257] On the other hand, when the value of the specific-signal-line flag 10f is "1", that indicates that a packet protocol is to be used. Hence, the process proceeds to step (4), where the data of the reception buffer storage 10b is moved to one of the printing-data buffer storage 10c and the control-command buffer storage 10d in accordance with information indicated in the packet header. At that time, only a portion other than the packet header is moved. Then, the interrupt processing

is terminated.

[0258] As described above, according to interrupt occurred every time data is input from the host apparatus 2, the input data is stored in the reception buffer storage 10b. Then, in accordance with the value of the specific-signal-line flag 10f or information in the packet header, the input data is stored in the printing-data buffer storage 10c or the control-command buffer storage 10d.

[0259] Next, a description will be provided of an outline of printing processing performed by the LBP 1 when input data from the host apparatus 2 is stored in the printing-data buffer storage 10c according to interrupt for storing the input data, with reference to FIG. 15.

[0260] When the CPU 7 has recognized that the input data has been stored in the printing-data buffer storage 10c, the CPU 7 interprets printing data stored in the printing-data buffer storage 10c by referring to a printing-data analysis program stored in a PROM 8, and converts the printing data into image data. The image data provided by the CPU 7 is stored in the frame buffer storage 10a.

[0261] The CPU 7 continues the processing by referring to the printing-data analysis program in the above-described manner. When the CPU 7 has found a sheet discharging command in the printing data, the CPU 7 transmits the image data stored in the frame buffer storage 10a to an engine 4 via an engine I/F 12. Upon reception of the image data transmitted from the CPU 7 via the engine I/F 12, the engine 4 provides printing output of the image data onto an actual recording medium.

[0262] As described above, when data has been input from the host apparatus 2, the LBP 1 performs printing processing, and image data is actually output onto a recording medium.

[0263] As described above, when a value "1" is stored in the specific-signal-line flag 10f, the printing apparatus operates using a packet protocol, and when a value "0" is stored in the specific-signal-line flag 10f, the printing apparatus operates using a conventional nonpacket protocol without using the packet protocol.

[0264] That is, by operating depending on whether or not the signal line when storing input data in the reception buffer storage 10b is the "High" state or the "Low" state, it is possible to operate by automatically determining whether or not the packet protocol is to be used.

[0265] The characteristic configuration of the third embodiment will now be described with reference to FIG. 18 and other figures.

[0266] A data processing method for a communication control apparatus (the controller 3) for performing two-way communication processing with a data processing apparatus (the host apparatus 2) via an interface having a predetermined number of communication lines configured in the above-described manner (in the third embodiment, a two-way communication interface conforming to the IEEE 1284.4, such as a Centronics interface), or a storage medium storing a program capable of

being read by a computer for controlling an apparatus for performing two-way communication processing with a data processing apparatus via an interface having a predetermined number of communication lines includes a storage step of storing data received from the data processing apparatus in one of regions of a memory capable of securing a plurality of memory regions for storing the received data (step (1) shown in FIG. 18), a detection step of detecting the state of a signal on a specific communication line from among the plurality of communication lines (step (2) shown in FIG. 18), a determination step of determining whether or not the type of the protocol for data communication indicates a packet protocol based on the result of the detection of the state of the signal on the specific signal line in the detection step (step (2) shown in FIG. 18), and a transfer step of transferring respective data to different regions in the memory by determining the type of data in the received data stored in some region of the memory when the determination step has determined that the type of the protocol indicates a packet protocol (step (3) or step (4) shown in FIG. 18). Hence, it is possible to automatically determine and set whether or not a specific packet protocol is to be used, and to freely construct a data communication environment capable of flexibly dealing with a request of packet communication.

[0267] Examples of the processing of step (3) and the processing of step (4) shown in FIG. 18 will now be described with reference to FIGS. 19 and 20.

[0268] FIGS. 19 and 20 are diagrams, each illustrating processing of moving received data in the communication control apparatus according to the third embodiment. FIG. 19 corresponds to data movement from the reception buffer storage 10b to the printing-data buffer storage 10c when the specific signal line assumes the "Low" state. FIG. 20 corresponds to data movement from the reception buffer storage 10b to the printing buffer storage 10c when the specific signal line assumes the "High" state.

[0269] When the result of the determination is negative in step (2) of the flowchart shown in FIG. 18, then, in step (3), the data string stored in the reception buffer storage 10b is moved to the printing-buffer storage 10c without modification.

[0270] On the other hand, when the result of the determination is affirmative in step (2) of the flowchart shown in FIG. 18, then, in step (4), a data string obtained by removing the 6-byte packet header from the data stored in the reception buffer storage 10b is moved to the printing-data buffer storage 10c or the control-command buffer storage 10d.

[0271] In this case, the location where data is to be moved and the size of data to be moved are determined from the information of the packet header. The data is determined to be moved to the printing buffer storage 10c based on the combination of Socket IDs (PSID = 0x10, SSID = 0x10) of the first byte 0x10 and the second byte 0x10 of the packet header.

[0272] It can be understood that the size of the packet including the packet header is 0x0010 bytes based on the third byte 0x00 and the fourth byte 0x10 of the packet header. Hence, the size of data other than the packet header is 10 bytes (0x0010 - 0x0006 = 0x000a) in decimal notation.

[0273] Accordingly, as shown in FIG. 20, 10-byte data obtained by removing the packet header from data stored in the reception buffer storage 10b is moved to the printing-data buffer storage 10c.

[0274] The third embodiment is particularly effective when the use of a packet protocol depends on whether a specific signal line assumes "High" or "Low", for example, in a case in which, when the host apparatus side transmits data using a conventional nonpacket protocol, the specific signal line is made "Low", and when the host apparatus side transmits data using a packet protocol, the specific signal line is made "High". This is because it is possible to normally operate by automatically determining whether or not a packet protocol is to be used.

[0275] In the third embodiment, the case of using a printer as an apparatus capable of communicating with a host apparatus has been described. However, the present invention may also be applied to any other computer peripheral apparatus, including an image input/output apparatus (a printer, a scanner device or the like), for performing data processing by communicating with a host apparatus, irrespective of the type of data processing.

[0276] In the third embodiment, the case of using an interface conforming to the IEEE 1284.4 has been described. However, the present invention may also be applied to an interface conforming to any other IEEE standards, a USB (Universal Serial Bus) or the like.

[0277] A description will now be provided of the configuration of a data processing program capable of being read by a printing system to which a communication control apparatus according to the present invention can be applied, with reference to the memory map shown in FIG. 21.

[0278] FIG. 21 is a diagram illustrating the memory map of a storage medium storing various data processing programs capable of being read by the communication control apparatus according to the present invention.

[0279] Although not illustrated, information for controlling a group of programs stored in the storage medium, comprising version information, authors and the like, is also stored. In addition, information depending on an OS (operating system) of the program reading side, comprising icons for identifying and displaying respective programs, and the like, may also be stored.

[0280] Furthermore, data belonging to various programs are also controlled by directory. Programs for installing various programs in the computer, programs for defrosting compressed programs to be installed, and the like are, in some cases, also stored.

[0281] The functions shown in FIGS. 12, 14 and 18 in the second and third embodiments may be executed by the host computer based on a program installed from the outside. In such a case, the present invention is also applied to a case in which a group of information including programs is supplied from a storage medium, such as CD(compact disc)-ROM, a flash memory, an FD (floppy disk) or the like, or from an external storage medium via a network to the output apparatus.

[0282] As described above, the objects of the present invention may, of course, also be achieved by supplying a system or an apparatus with a storage medium storing program codes of software for realizing the functions of the above-described embodiments, and reading and executing the program codes stored in the storage medium by means of a computer (or a CPU or an MPU (microprocessor unit)) of the system or the apparatus.

[0283] In such a case, the program codes themselves read from the storage medium realize the new functions of the present invention, so that the storage medium storing the program codes constitutes the present invention.

[0284] For example, a floppy disk, a hard disk, an optical disk, a magnetooptical disk, a CD-ROM, a CD-R(recordable), a magnetic tape, a nonvolatile memory card, a ROM, an EEPROM(electrically erasable and programmable)ROM or the like may be used as the storage medium for supplying the program codes.

[0285] The present invention may, of course, be applied not only to a case in which the functions of the above-described embodiments are realized by executing program codes read by a computer, but also to a case in which an OS or the like operating in a computer executes a part or the entirety of actual processing, and the functions of the above-described embodiments are realized by the processing.

[0286] The present invention may, of course, be applied to a case in which, after writing program codes read from a storage medium into a memory provided in a function expanding board inserted into a computer or in a function expanding unit connected to the computer, a CPU or the like provided in the function expanding board or the function expanding unit performs a part or the entirety of actual processing, and the functions of the above-described embodiments are realized by the processing.

[0287] The individual components shown in outline or designated by blocks in the drawings are all well known in the interface control apparatus arts and their specific construction and operation are not critical to the operation or the best mode for carrying out the invention.

[0288] While the present invention has been described with respect to what are presently considered to be the preferred embodiments, it is to be understood that the invention is not limited to the disclosed embodiments. To the contrary, the present invention is intended to cover various modifications and equivalent arrangements included within the spirit and scope of the

appended claims. The scope of the following claims is to be accorded the broadest interpretation so as to encompass all such modifications and equivalent structures and functions.

#### Claims

1. A communication control apparatus for performing two-way communication processing with a data processing apparatus via a predetermined interface, said communication control apparatus comprising:

storage means having a first area and a second area for storing data received from the data processing apparatus; and control means for storing, when packet data has been received from the data processing apparatus, the packet data in the first area or the second area in accordance with a type of data in order to use a packet protocol, and for storing, when nonpacket data has been received from the data processing apparatus, the nonpacket data in the first area in order to use a nonpacket protocol.

2. A communication control apparatus according to Claim 1, further comprising first setting means for setting in advance whether or not the packet protocol is to be used.
3. A communication control apparatus according to Claim 1, further comprising second setting means for setting in advance whether or not the packet protocol is to be used for respective communication modes.
4. A communication control apparatus according to Claim 3, wherein the communication modes include a first data communication mode in which data is communicated from the data processing apparatus to an apparatus including said communication control apparatus, and a second data communication mode in which data is communicated from the data processing apparatus to the apparatus including said communication control apparatus, or vice versa.
5. A communication control apparatus according to Claim 1, further comprising detection means for detecting a state of a signal on each of a plurality of communication lines of the interface, wherein said control means determines whether or not the packet protocol is to be used in accordance with the state of the signal.
6. A communication control method in a communication control apparatus for performing two-way com-

munication processing with a data processing apparatus via a predetermined interface, said method comprising the steps of:

storing, when packet data has been received 5  
from the data processing apparatus, the packet  
data in a first area or a second area of a mem-  
ory in accordance with a type of data in order to  
use a packet protocol; and  
storing, when nonpacket data has been 10  
received from the data processing apparatus,  
the nonpacket data in the first area in order to  
use a nonpacket protocol.

7. A method according to Claim 6, further comprising 15  
a first setting step of setting in advance whether or  
not the packet protocol is to be used.
8. A method according to Claim 6, further comprising  
a second setting step of setting in advance whether 20  
or not the packet protocol is to be used for respec-  
tive communication modes.
9. A method according to Claim 8, wherein the com- 25  
munication modes includes a first data communica-  
tion mode in which data is communicated from the  
data processing apparatus to an apparatus includ-  
ing the communication control apparatus, and a  
second data communication mode in which data is 30  
communicated from the data processing apparatus  
to the apparatus including the communication con-  
trol apparatus, or vice versa.
10. A method according to Claim 6, further comprising 35  
a detection step of detecting a state of a signal on  
each of a plurality of communication lines of the  
interface, wherein it is determined whether or not  
the packet protocol is to be used in accordance with  
the state of the signal. 40
11. A storage medium storing a control program to be 45  
used in a communication control apparatus for per-  
forming two-way communication processing with a  
data processing apparatus via a predetermined  
interface, said storage medium comprising:

a code of storing, when packet data has been  
received from the data processing apparatus,  
the packet data in a first area or a second area 50  
of a memory in accordance with a type of data  
in order to use a packet protocol; and  
a code of storing, when nonpacket data has  
been received from the data processing appa-  
ratus, the nonpacket data in the first area in  
order to use a nonpacket protocol. 55

FIG.1

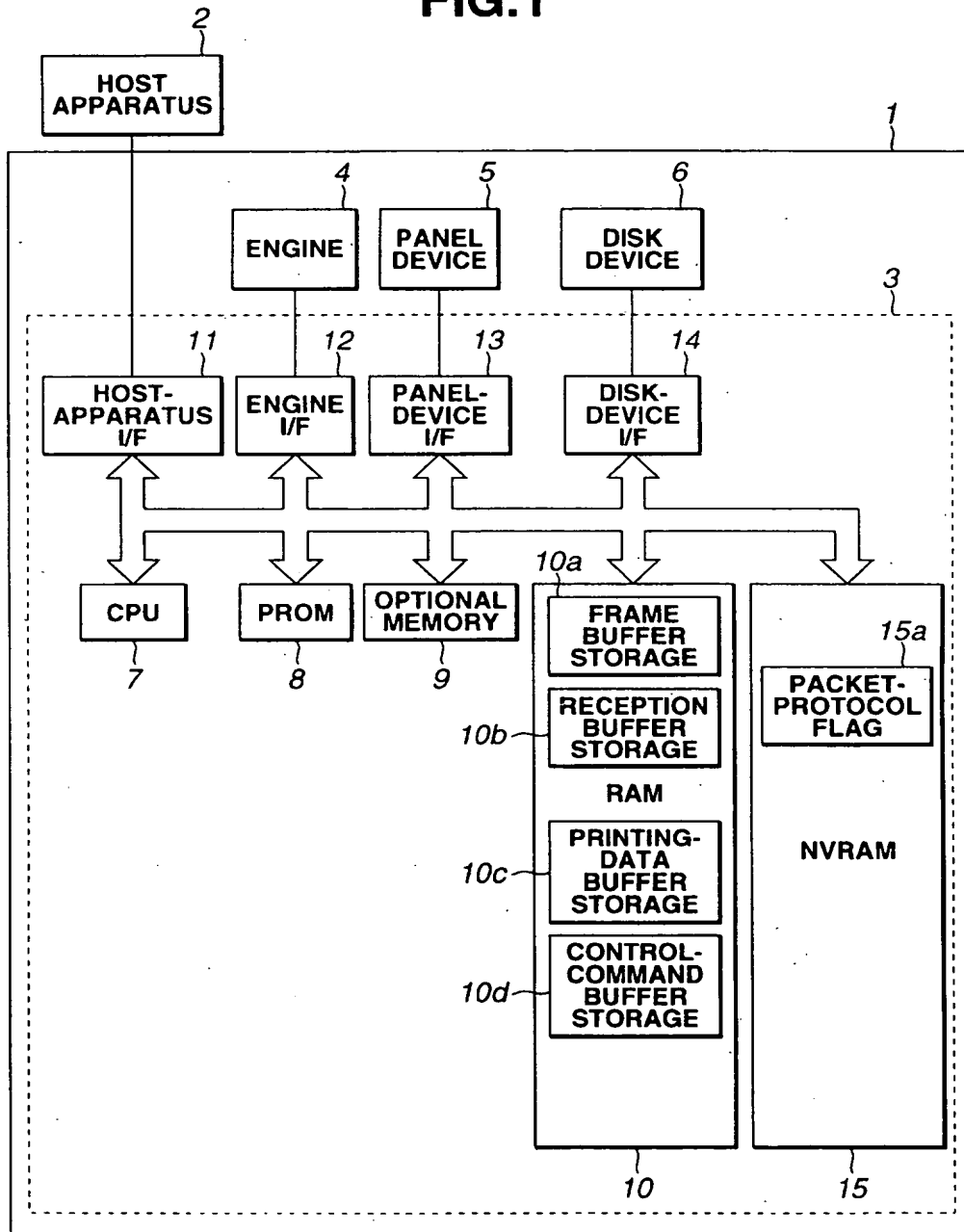
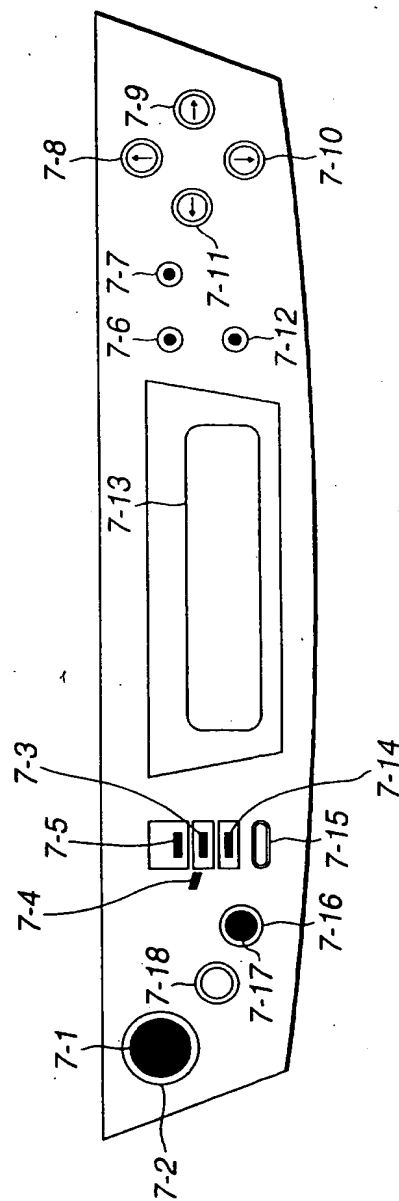


FIG.2





**FIG.3**

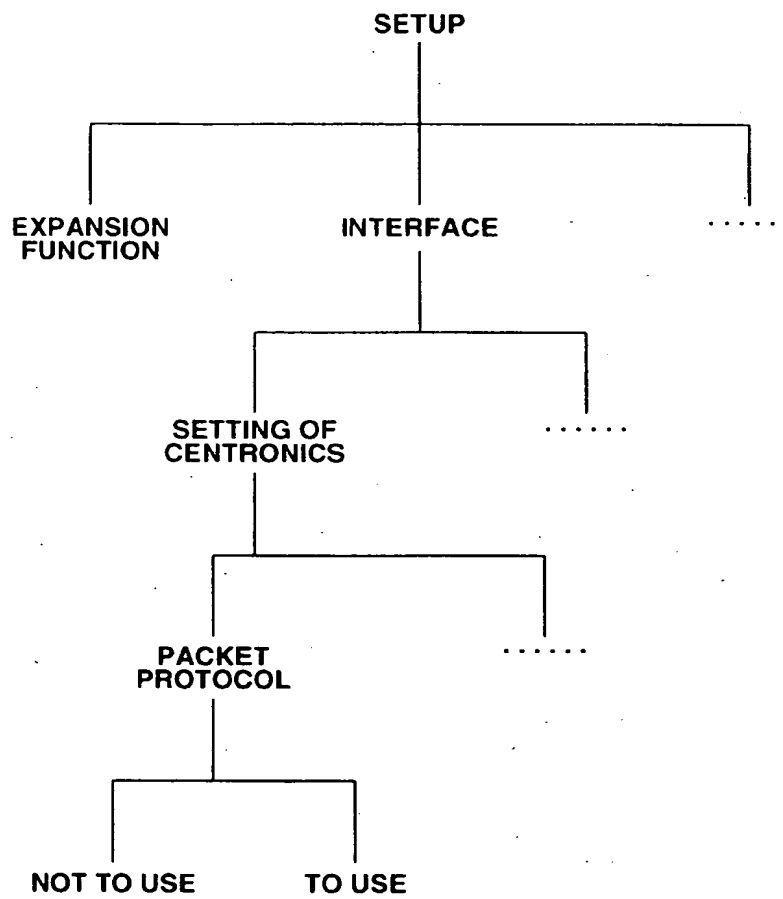


FIG.4

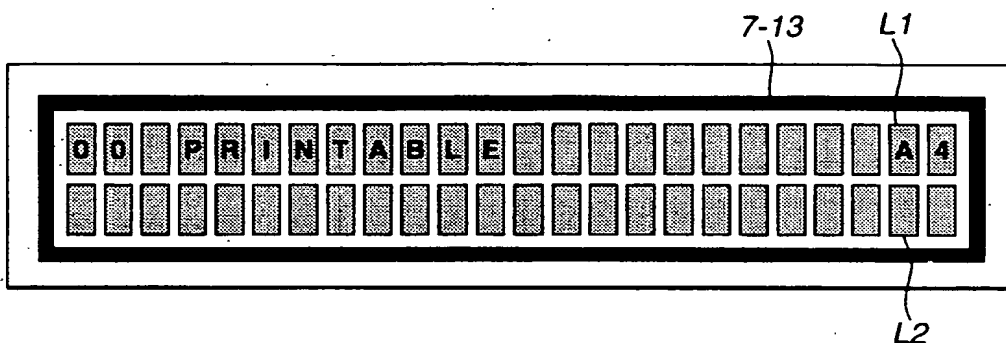


FIG.5

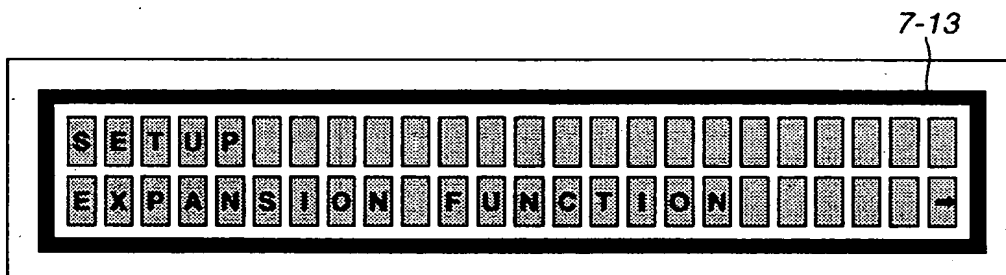
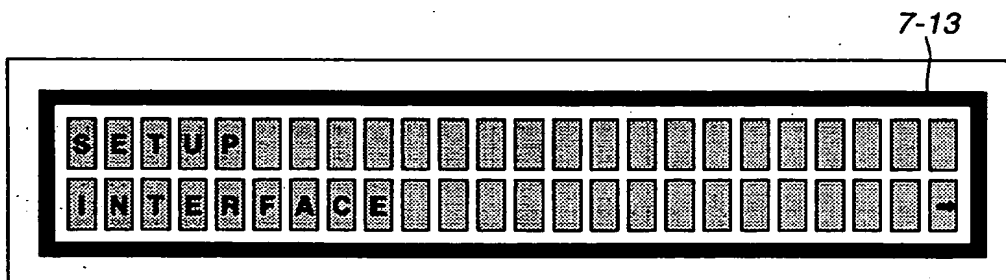


FIG.6



7-13



7-13



**7-13**



FIG.10

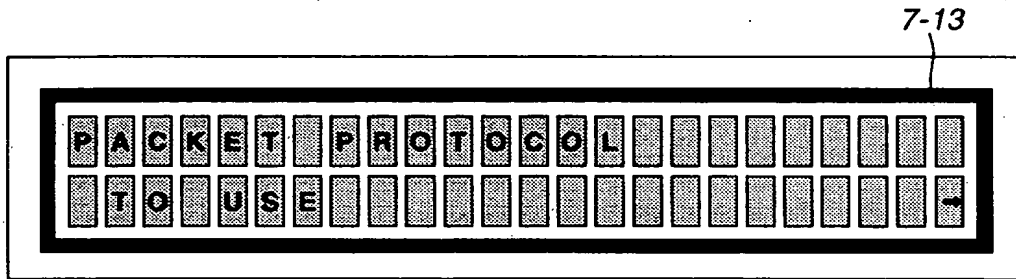
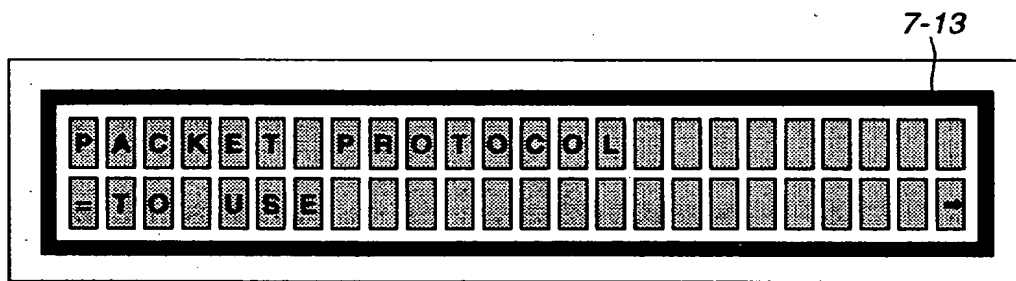


FIG.11



**FIG.12**

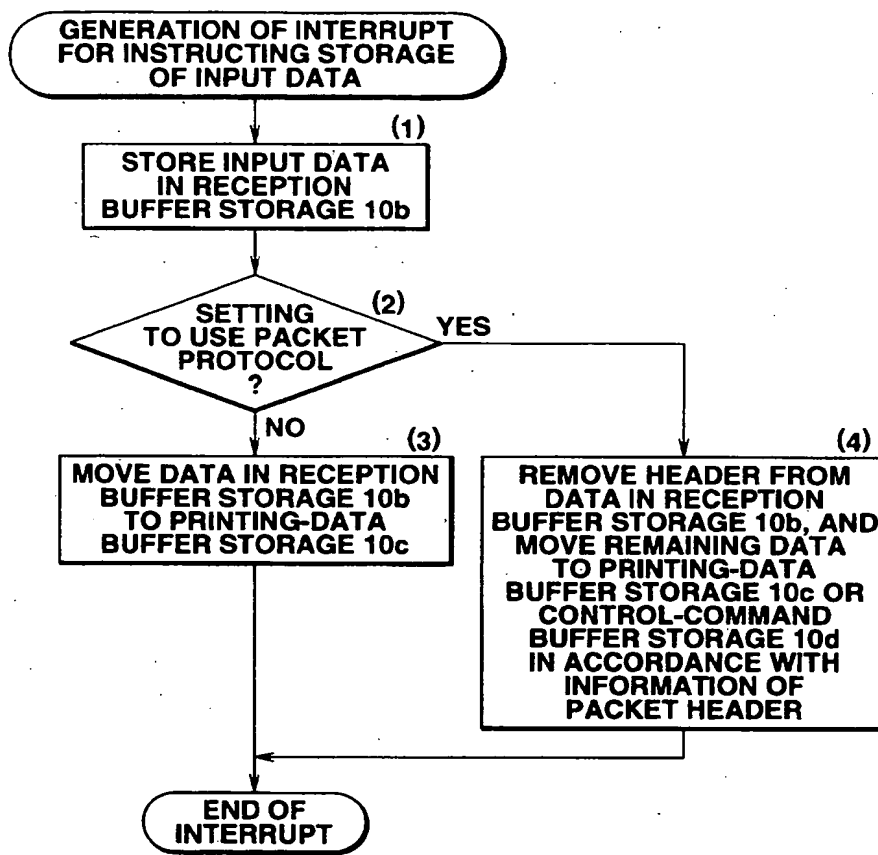


FIG.13

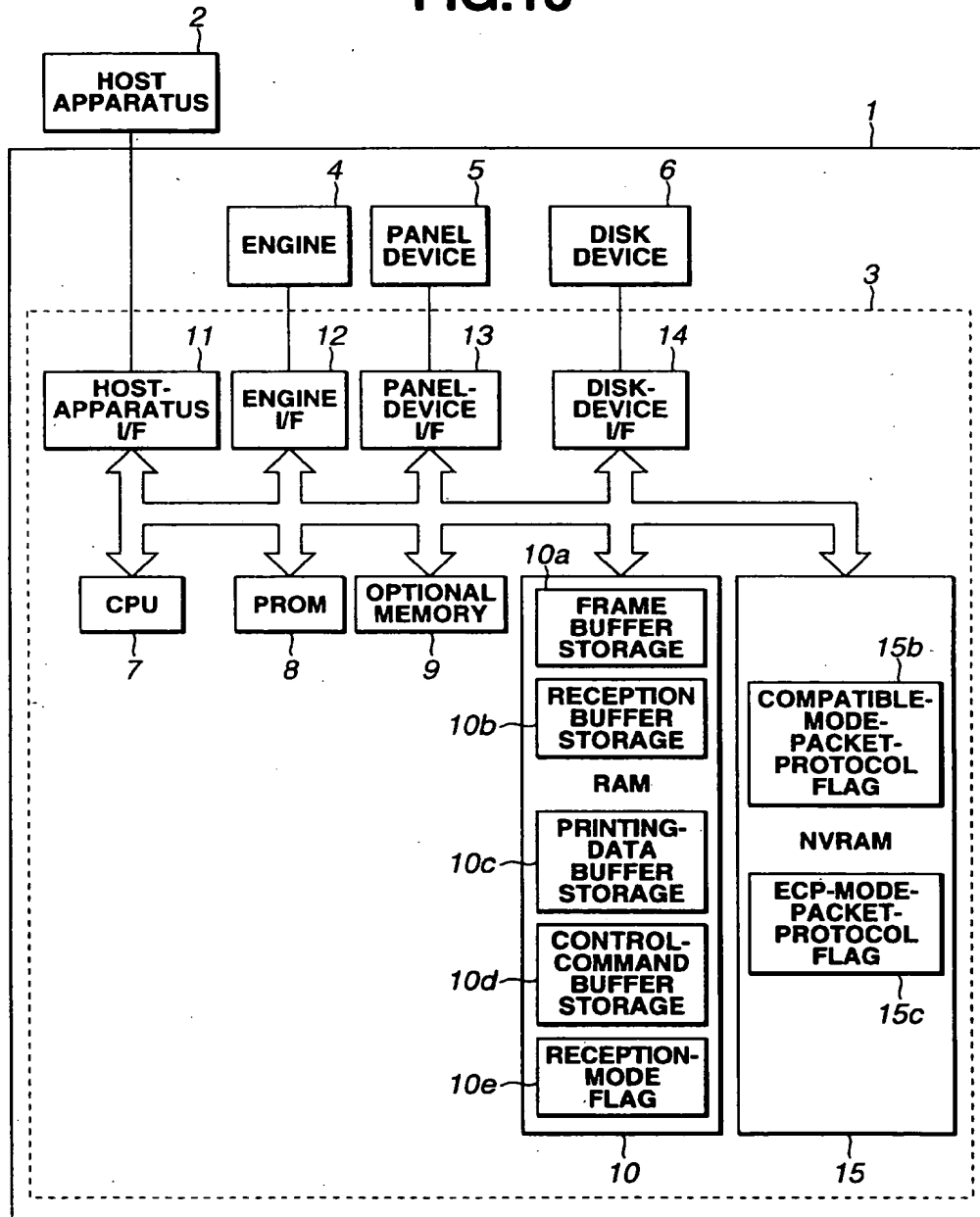


FIG.14

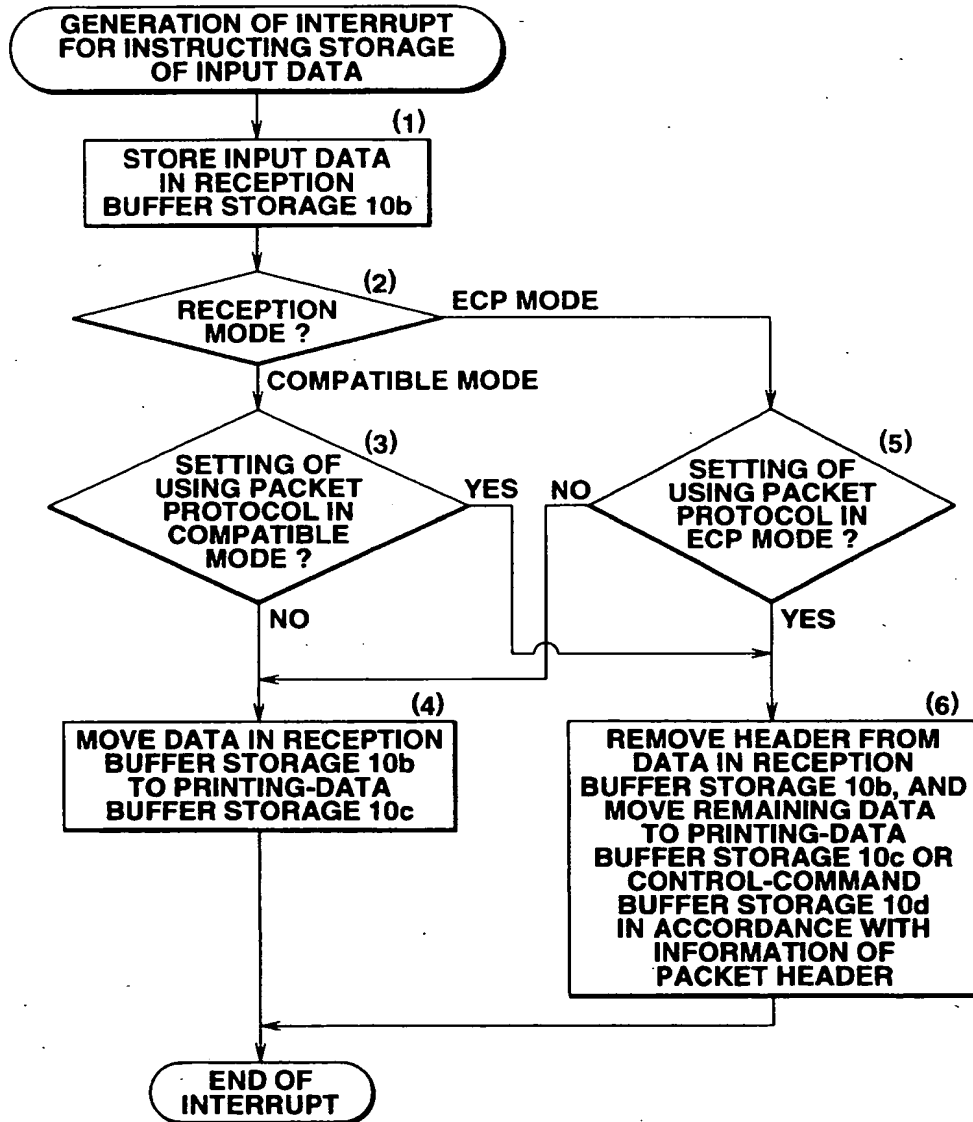
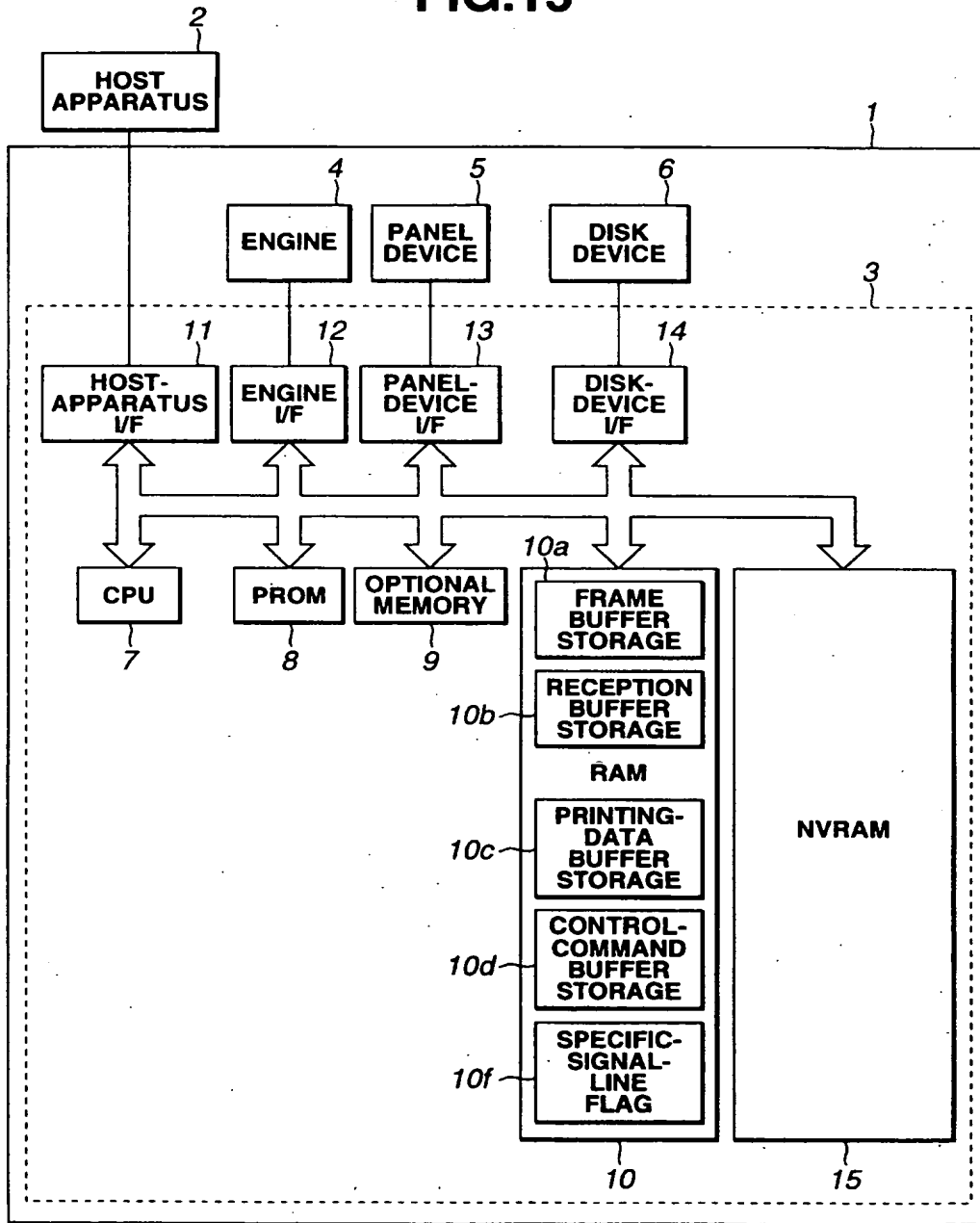
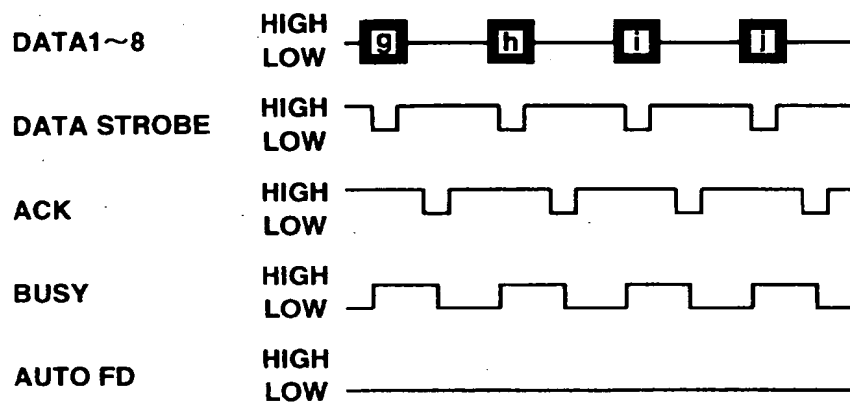


FIG.15





**FIG.16**



**FIG.17**

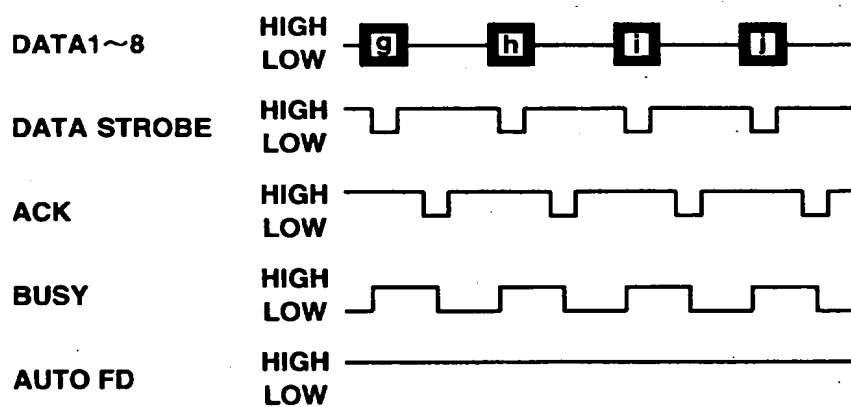
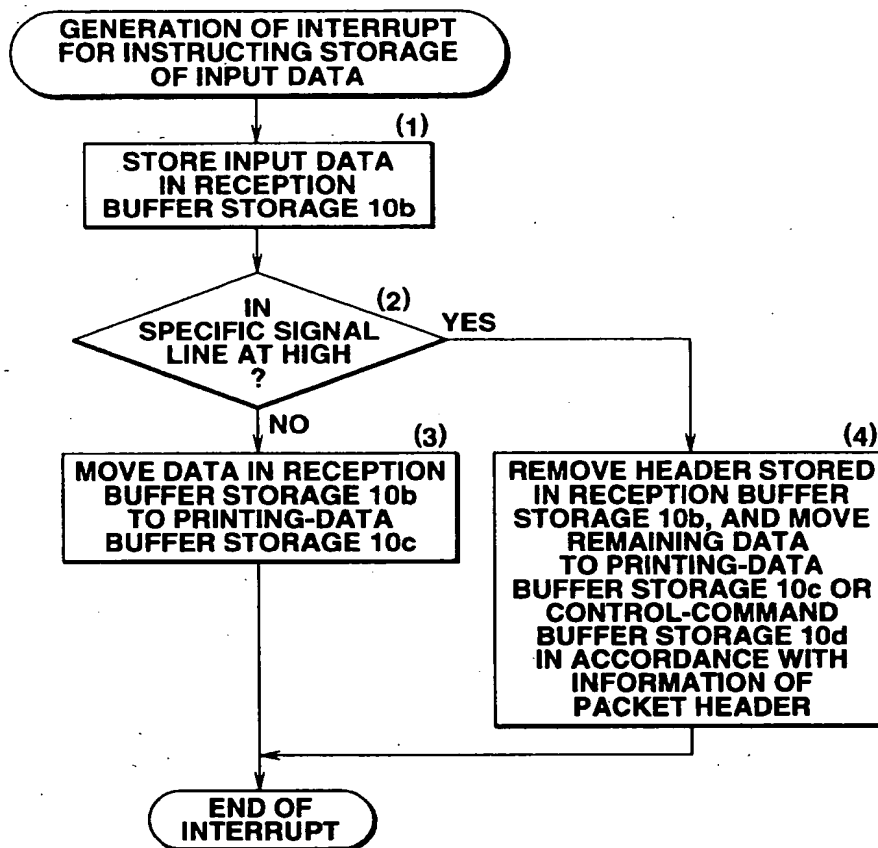
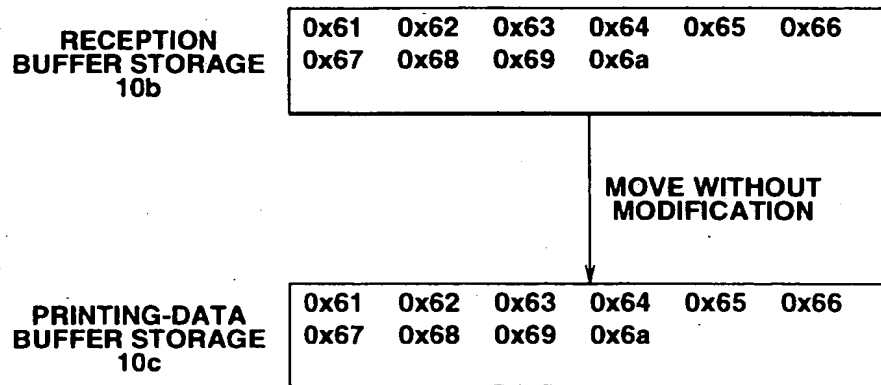


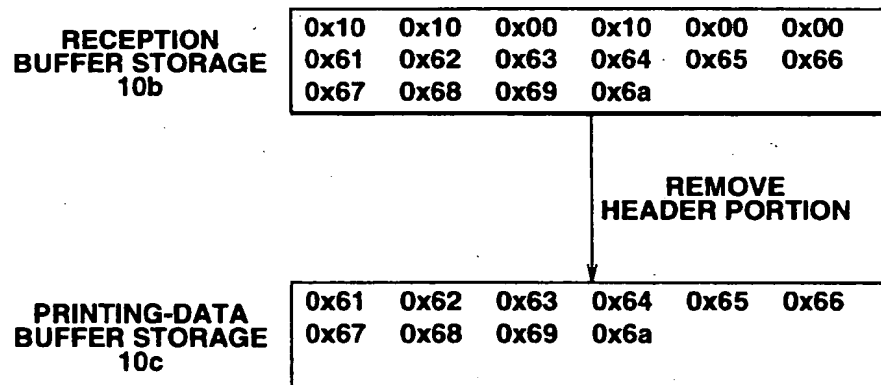
FIG.18



**FIG.19**



**FIG.20**

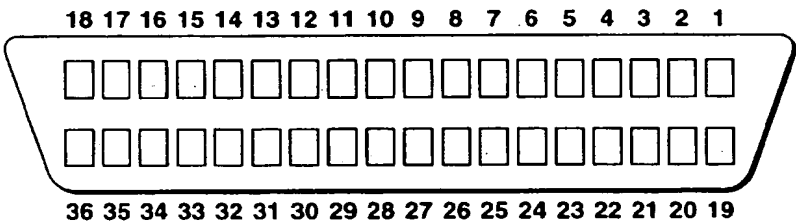


## FIG.21

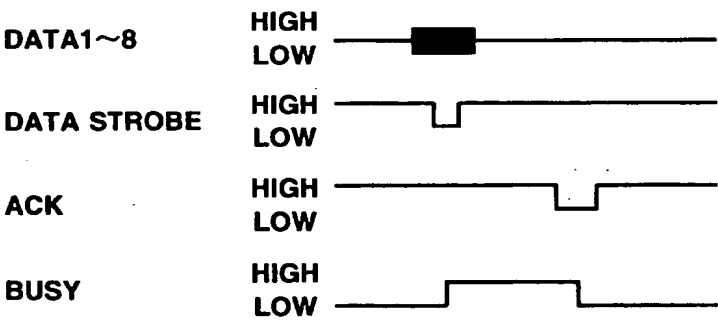
STORAGE MEDIUM SUCH AS FD,  
CD-ROM OR THE LIKE

DIRECTORY INFORMATION
FIRST DATA PROCESSING PROGRAM PROGRAM-CODE GROUP CORRESPONDING TO STEPS OF FLOWCHART SHOWN IN FIG.12
SECOND DATA PROCESSING PROGRAM PROGRAM-CODE GROUP CORRESPONDING TO STEPS OF FLOWCHART SHOWN IN FIG.14
THIRD DATA PROCESSING PROGRAM PROGRAM-CODE GROUP CORRESPONDING TO STEPS OF FLOWCHART SHOWN IN FIG.18

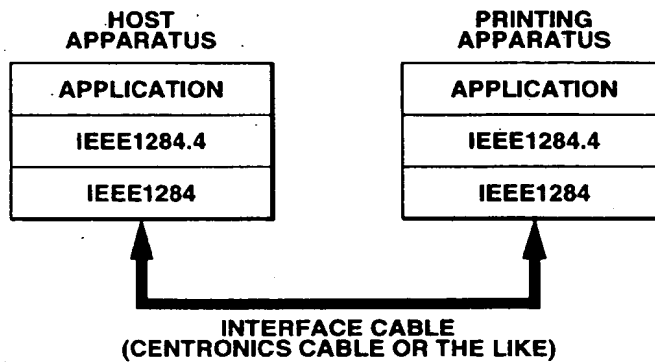
**FIG.22**  
**PRIOR ART**



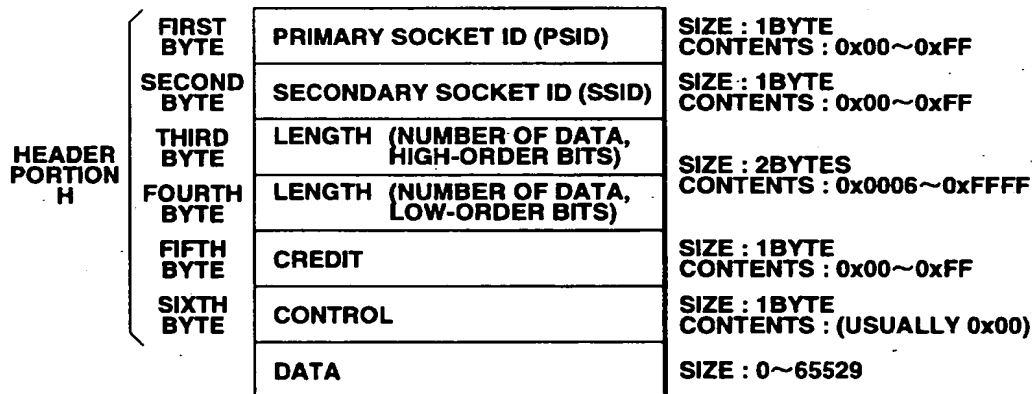
**FIG.23**  
**PRIOR ART**



**FIG.24**  
**PRIOR ART**



**FIG.25**  
**PRIOR ART**



## FIG.26A PRIOR ART

DATA IN THE FORM OF PACKET

16-BYTE GROUP BASED ON  
INFORMATION REPRESENTED  
BY HEADER (6 BYTES) AT HEAD

PRIMARY SOCKET ID (PSID)
SECONDARY SOCKET ID (SSID)
LENGTH (HIGH-ORDER BIT = 0x00)
LENGTH (LOW-ORDER BIT 0x10)
CREDIT
CONTROL
a
b
c
d
e
f
g
h
i
j

## FIG.26B PRIOR ART

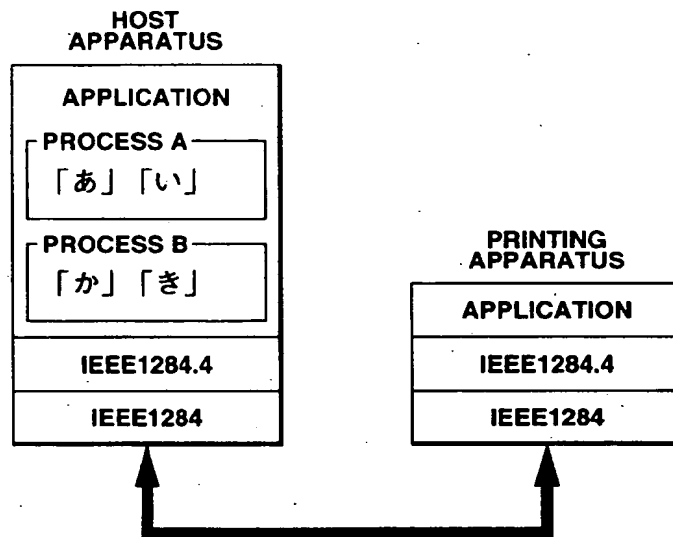
DATA IN THE FORM OF PACKET

NO CONNECTION BETWEEN  
RESPECTIVE DATA

a
b
c
d
e
f
g
h
i
j

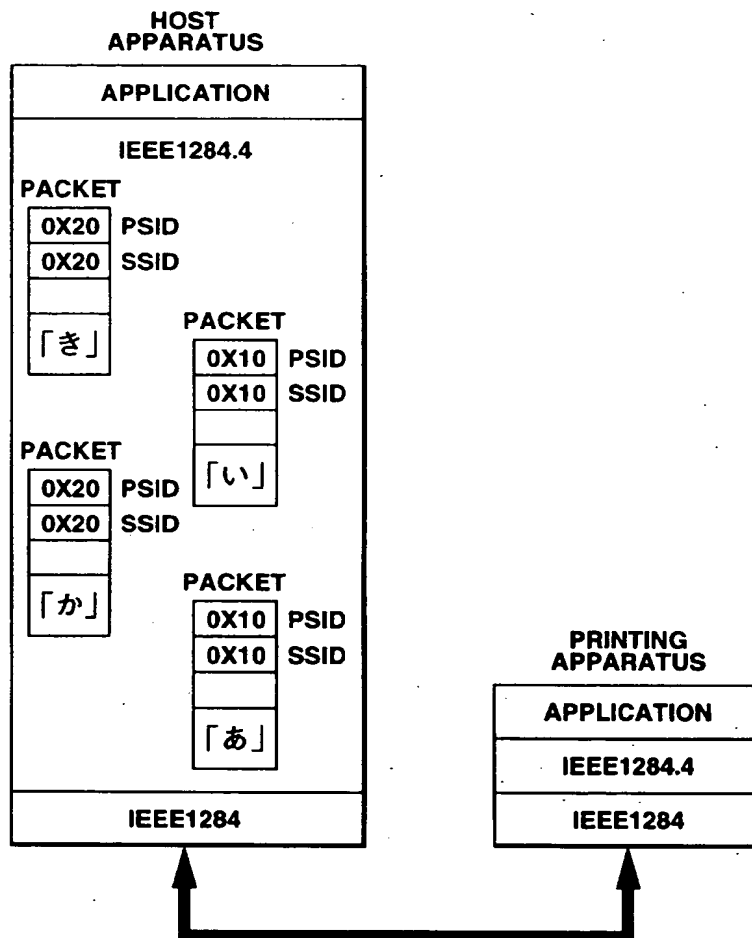
WHEN TRANSFERRING DATA  
COMPRISING 10 CHARACTERS  
(SUPPOSED TO BE 10 BYTES)  
abcdefghijkl

**FIG.27**  
**PRIOR ART**

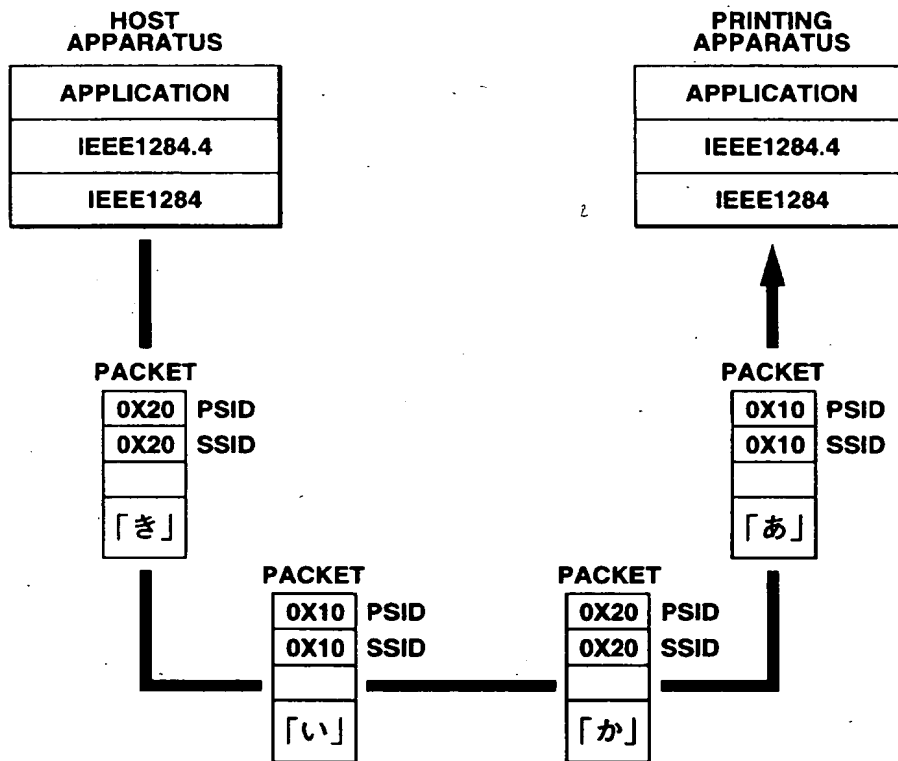




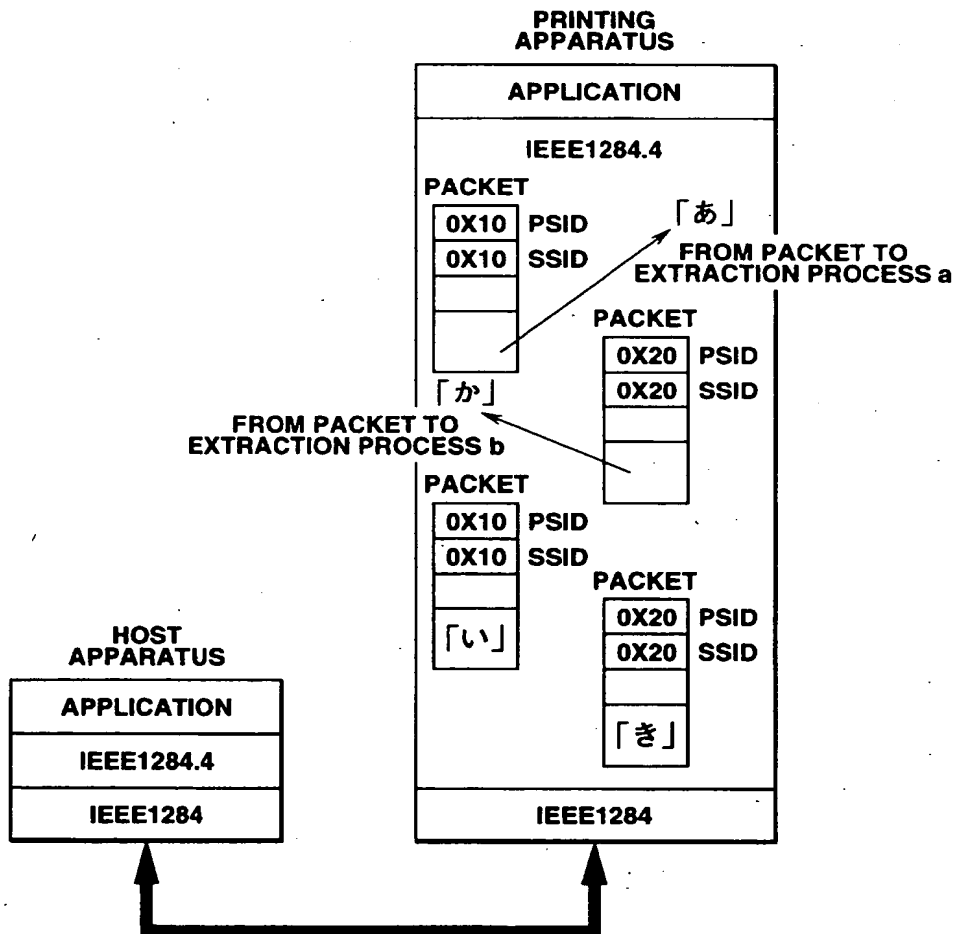
**FIG.28**  
**PRIOR ART**



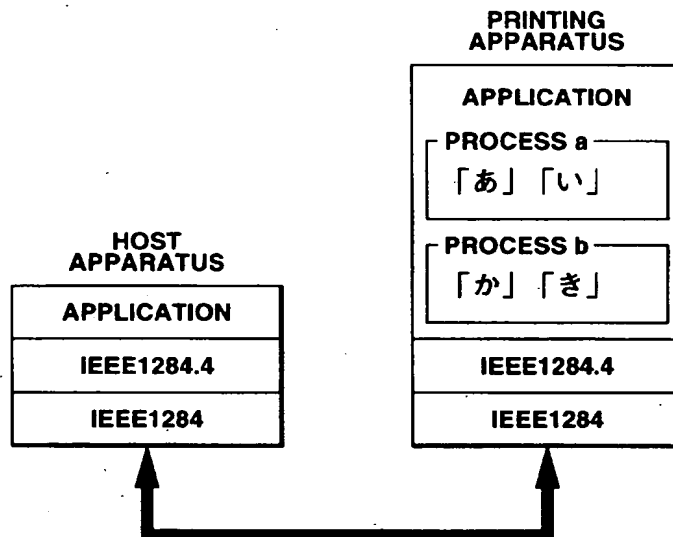
**FIG.29**  
**PRIOR ART**



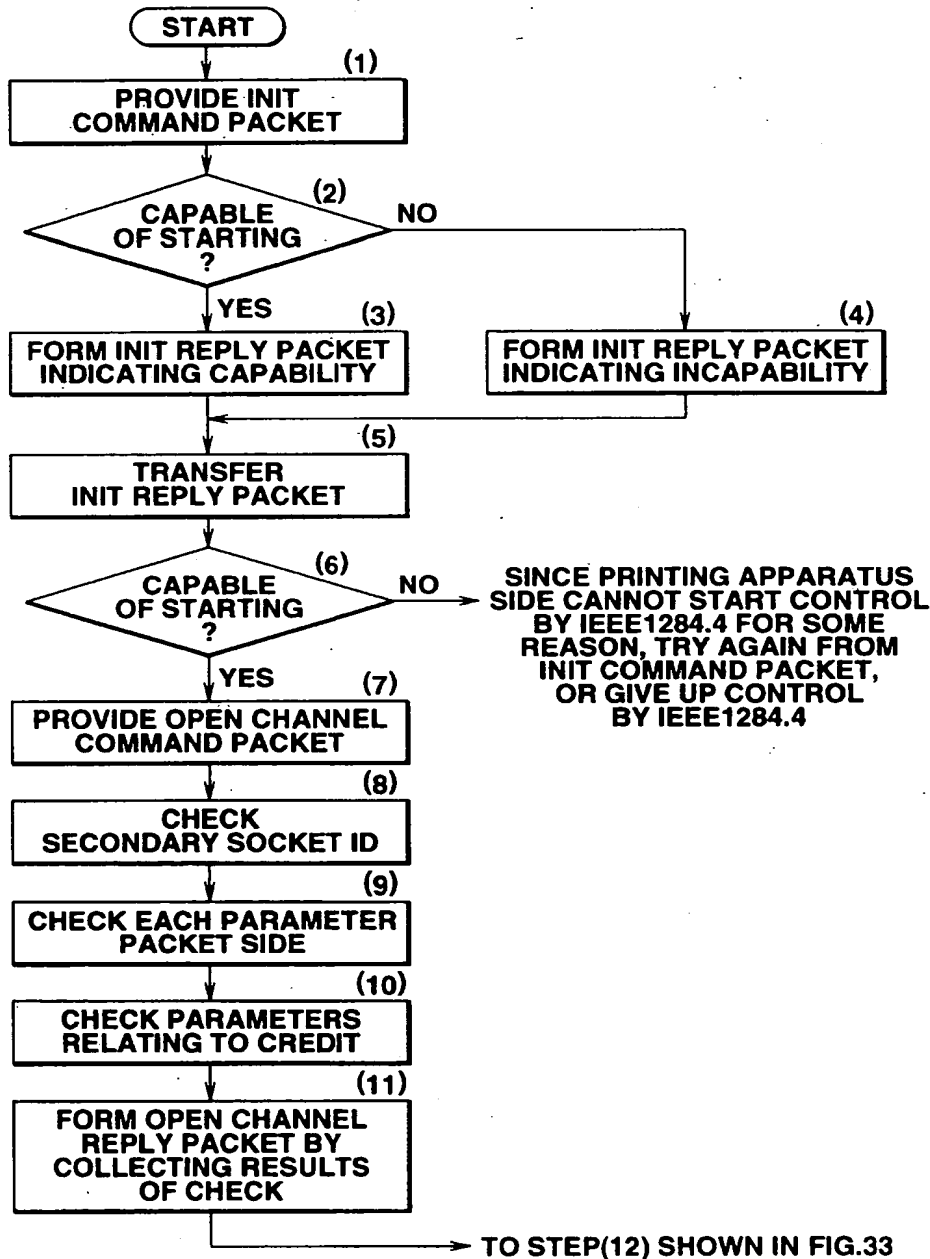
**FIG.30**  
**PRIOR ART**



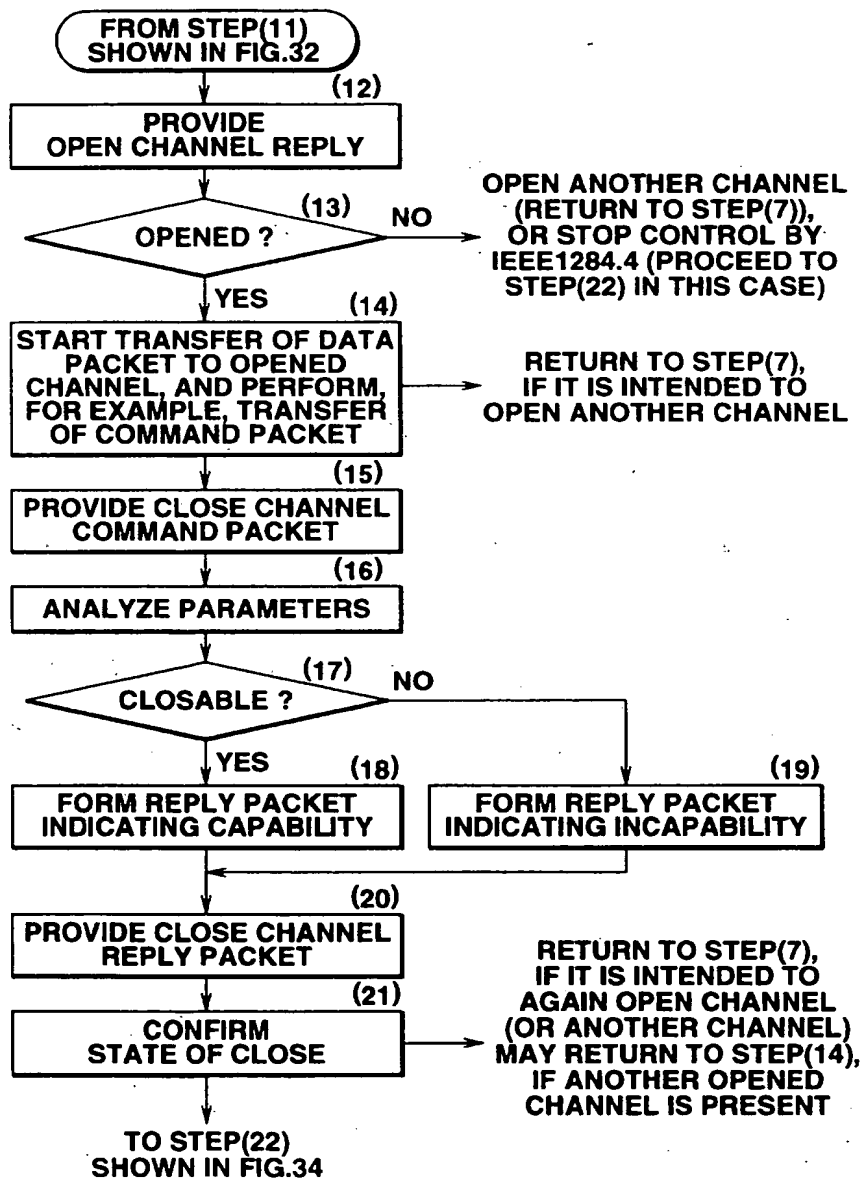
**FIG.31**  
**PRIOR ART**



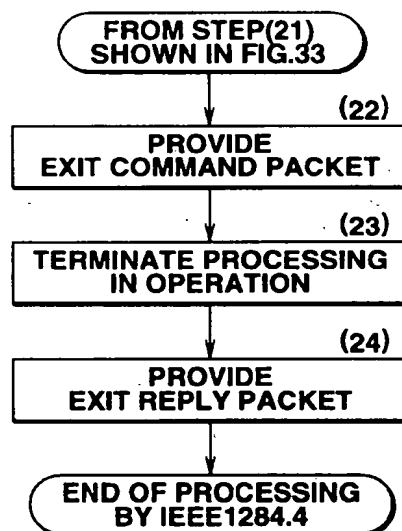
# FIG.32 PRIOR ART



# **FIG.33** **PRIOR ART**



# **FIG.34** **PRIOR ART**



# **FIG.35** **PRIOR ART**

HEADER	FIRST BYTE	PRIMARY SOCKET ID (PSID)	CONTENTS : 0x00
	SECOND BYTE	SECONDARY SOCKET ID (SSID)	CONTENTS : 0x00
	THIRD BYTE	LENGTH (NUMBER OF DATA, HIGH-ORDER BITS)	CONTENTS : 0x0008 (WITH 2 BYTES)
	FOURTH BYTE	LENGTH (NUMBER OF DATA, LOW-ORDER BITS)	
	FIFTH BYTE	CREDIT	CONTENTS : 0x00~0xFF
	SIXTH BYTE	CONTROL	CONTENTS : 0x00
	SEVENTH BYTE	COMMAND	CONTENTS : 0x00
	EIGHTH BYTE	REVISION	CONTENTS : 0x10

# **FIG.36** **PRIOR ART**

HEADER	FIRST BYTE	PRIMARY SOCKET ID (PSID)	CONTENTS : 0x00
	SECOND BYTE	SECONDARY SOCKET ID (SSID)	CONTENTS : 0x00
	THIRD BYTE	LENGTH (NUMBER OF DATA, HIGH-ORDER BITS)	CONTENTS : 0x0009 (WITH 2 BYTES)
	FOURTH BYTE	LENGTH (NUMBER OF DATA, LOW-ORDER BITS)	
	FIFTH BYTE	CREDIT	CONTENTS : 0x00~0xFF
	SIXTH BYTE	CONTROL	CONTENTS : 0x00
	SEVENTH BYTE	COMMAND	CONTENTS : 0x80
	EIGHTH BYTE	RESULT	CONTENTS : 0x00 (IN NORMAL STATE)
	NINTH BYTE	REVISION	CONTENTS : 0x10



# FIG.37

## PRIOR ART

HEADER	FIRST BYTE	PSID	CONTENTS : 0x00
	SECOND BYTE	SSID	CONTENTS : 0x00
	THIRD BYTE	LENGTH (NUMBER OF DATA, HIGH-ORDER BITS)	CONTENTS : 0x0011 (WITH 2 BYTES)
	FOURTH BYTE	LENGTH (NUMBER OF DATA, LOW-ORDER BITS)	
	FIFTH BYTE	CREDIT	CONTENTS : 0x00~0xFF
	SIXTH BYTE	CONTROL	CONTENTS : 0x00
	SEVENTH BYTE	COMMAND	CONTENTS : 0x01
	EIGHTH BYTE	PRIMARY SOCKET ID	CONTENTS : 0x01~0xFF
	NINTH BYTE	SECONDARY SOCKET ID	CONTENTS : 0x01~0xFF
	TENTH BYTE	PRIMARY TO SECONDARY PACKET SIZE HIGH-ORDER BITS	CONTENTS : 0x0000 OR 0x0006~0xFFFF
	ELEVENTH BYTE	PRIMARY TO SECONDARY PACKET SIZE LOW-ORDER BITS	
	TWELFTH BYTE	SECONDARY TO PRIMARY PACKET SIZE HIGH-ORDER BITS	CONTENTS : 0x0000 OR 0x0006~0xFFFF
	THIRTEENTH BYTE	SECONDARY TO PRIMARY PACKET SIZE LOW-ORDER BITS	
	FOURTEENTH BYTE	CREDIT REQUESTED (HIGH-ORDER BITS)	CONTENTS : 0x0000 ~0xFFFF
	FIFTEENTH BYTE	CREDIT REQUESTED (LOW-ORDER BITS)	
	SIXTEENTH BYTE	MAXIMUM OUTSTANDING CREDIT (HIGH-ORDER BITS)	CONTENTS : 0x0000 ~0xFFFF
	SEVENTEENTH BYTE	MAXIMUM OUTSTANDING CREDIT (LOW-ORDER BITS)	

# FIG.38

## PRIOR ART

HEADER	FIRST BYTE	PSID	CONTENTS : 0x00
	SECOND BYTE	SSID	CONTENTS : 0x00
	THIRD BYTE	LENGTH (NUMBER OF DATA, HIGH-ORDER BITS)	CONTENTS : 0x0010
	FOURTH BYTE	LENGTH (NUMBER OF DATA, LOW-ORDER BITS)	(WITH 2 BYTES)
	FIFTH BYTE	CREDIT	CONTENTS : 0x00~0xFF
	SIXTH BYTE	CONTROL	CONTENTS : 0x00
	SEVENTH BYTE	COMMAND	CONTENTS : 0x81
	EIGHTH BYTE	RESULT	CONTENTS : 0x00
	NINTH BYTE	PRIMARY SOCKET ID	(IN NORMAL STATE)
	TENTH BYTE	SECONDARY SOCKET ID	CONTENTS : 0x01~0xFF
	ELEVENTH BYTE	PRIMARY TO SECONDARY PACKET SIZE HIGH-ORDER BITS	CONTENTS : 0x01~0xFF
	TWELFTH BYTE	PRIMARY TO SECONDARY PACKET SIZE LOW-ORDER BITS	CONTENTS : 0x0000 OR 0x0006~0xFFFF
	THIRTEENTH BYTE	SECONDARY TO PRIMARY PACKET SIZE HIGH-ORDER BITS	CONTENTS : 0x0000 OR 0x0006~0xFFFF
	FOURTEENTH BYTE	SECONDARY TO PRIMARY PACKET SIZE LOW-ORDER BITS	CONTENTS : 0x0000 OR 0x0006~0xFFFF
	FIFTEENTH BYTE	CREDIT (HIGH-ORDER BITS)	CONTENTS : 0x0000
	SIXTEENTH BYTE	CREDIT (LOW-ORDER BITS)	~0xFFFF

# **FIG.39** **PRIOR ART**

HEADER	FIRST BYTE	PRIMARY SOCKET ID (PSID)	CONTENTS : 0x00
	SECOND BYTE	SECONDARY SOCKET ID (SSID)	CONTENTS : 0x00
	THIRD BYTE	LENGTH (NUMBER OF DATA, HIGH-ORDER BITS)	CONTENTS : 0x0009 (WITH 2 BYTES)
	FOURTH BYTE	LENGTH (NUMBER OF DATA, LOW-ORDER BITS)	
	FIFTH BYTE	CREDIT	CONTENTS : 0x00~0xFF
	SIXTH BYTE	CONTROL	CONTENTS : 0x00
	SEVENTH BYTE	COMMAND	CONTENTS : 0x02
	EIGHTH BYTE	PRIMARY SOCKET ID	CONTENTS : 0x01~FF
	NINTH BYTE	SECONDARY SOCKET ID	CONTENTS : 0x01~FF

# **FIG.40** **PRIOR ART**

HEADER	FIRST BYTE	PRIMARY SOCKET ID (PSID)	CONTENTS : 0x00
	SECOND BYTE	SECONDARY SOCKET ID (SSID)	CONTENTS : 0x00
	THIRD BYTE	LENGTH (NUMBER OF DATA, HIGH-ORDER BITS)	CONTENTS : 0x000A (WITH 2 BYTES)
	FOURTH BYTE	LENGTH (NUMBER OF DATA, LOW-ORDER BITS)	
	FIFTH BYTE	CREDIT	CONTENTS : 0x00~0xFF
	SIXTH BYTE	CONTROL	CONTENTS : 0x00
	SEVENTH BYTE	COMMAND	CONTENTS : 0x82
	EIGHTH BYTE	RESULT	CONTENTS : 0x00 (IN NORMAL STATE)
	NINTH BYTE	PRIMARY SOCKET ID	CONTENTS : 0x01~FF
	TENTH BYTE	SECONDARY SOCKET ID	CONTENTS : 0x01~FF

# **FIG.41** **PRIOR ART**

HEADER	FIRST BYTE	PRIMARY SOCKET ID (PSID)	CONTENTS : 0x00
	SECOND BYTE	SECONDARY SOCKET ID (SSID)	CONTENTS : 0x00
	THIRD BYTE	LENGTH (NUMBER OF DATA, HIGH-ORDER BITS)	CONTENTS : 0x0007 (WITH 2 BYTES)
	FOURTH BYTE	LENGTH (NUMBER OF DATA, LOW-ORDER BITS)	
	FIFTH BYTE	CREDIT	CONTENTS : 0x00~0xFF
	SIXTH BYTE	CONTROL	CONTENTS : 0x00
	SEVENTH BYTE	COMMAND	CONTENTS : 0x08

# **FIG.42** **PRIOR ART**

HEADER	FIRST BYTE	PRIMARY SOCKET ID (PSID)	CONTENTS : 0x00
	SECOND BYTE	SECONDARY SOCKET ID (SSID)	CONTENTS : 0x00
	THIRD BYTE	LENGTH (NUMBER OF DATA, HIGH-ORDER BITS)	CONTENTS : 0x0008 (WITH 2 BYTES)
	FOURTH BYTE	LENGTH (NUMBER OF DATA, LOW-ORDER BITS)	
	FIFTH BYTE	CREDIT	CONTENTS : 0x00~0xFF
	SIXTH BYTE	CONTROL	CONTENTS : 0x00
	SEVENTH BYTE	COMMAND	CONTENTS : 0x88
	EIGHTH BYTE	RESULT	CONTENTS : 0x00 (IN NORMAL STATE)

# **FIG.43** **PRIOR ART**

HOST APPARATUS			PRINTING APPARATUS	
CREDIT IN DATA TRANSFER CHANNEL	CREDIT IN COMMAND CHANNEL		CREDIT IN COMMAND CHANNEL	CREDIT IN DATA TRANSFER CHANNEL
0	2		2	0
0	-1 = 1	INIT →	+1 = 3	0
0	+1 = 2	← INIT REPLY	-1 = 2	0
0	-1 = 1	OPEN CHANNEL →	+1 = 3	0
+2 = 2	+1 = 2	PROVIDE TWO PARAMETERS CREDIT OPEN CHANNEL REPLY ←	-1 = 2	0
-1 = 2	2	TRANSFER ONE DATA PACKET →	2	0
-1 = 0	2	PROVIDE ONE DATA PACKET AND ONE CREDIT WITH TRANSFER PARAMETER →	2	+1 = 1
0	-1 = 1	CREDIT REQUEST →	+1 = 3	1
+1 = 1	+1 = 2	PROVIDE ONE CREDIT WITH PARAMETER OPEN REQUEST REPLY ←	-1 = 2	1
+1 = 2	2	PROVIDE ONE DATA PACKET AND ONE CREDIT WITH TRANSFER PARAMETER ←	2	-1 = 0

# FIG.44 PRIOR ART

HEADER	FIRST BYTE	PSID	CONTENTS : 0x00
	SECOND BYTE	SSID	CONTENTS : 0x00
	THIRD BYTE	LENGTH (NUMBER OF DATA, HIGH-ORDER BITS)	CONTENTS : 0x000D (WITH 2 BYTES)
	FOURTH BYTE	LENGTH (NUMBER OF DATA, LOW-ORDER BITS)	
	FIFTH BYTE	CREDIT	CONTENTS : 0x00~0xFF
	SIXTH BYTE	CONTROL	CONTENTS : 0x00
	SEVENTH BYTE	COMMAND	CONTENTS : 0x04
	EIGHTH BYTE	PRIMARY SOCKET ID	CONTENTS : 0x01~0xFF
	NINTH BYTE	SECONDARY SOCKET ID	CONTENTS : 0x01~0xFF
	TENTH BYTE	CREDIT REQUESTED (HIGH-ORDER BITS)	CONTENTS : 0x0000 ~0xFFFF
	ELEVENTH BYTE	CREDIT REQUESTED (LOW-ORDER BITS)	
	TWELFTH BYTE	MAXIMUM OUTSTANDING CREDIT (HIGH-ORDER BITS)	CONTENTS : 0x0000 ~0xFFFF
	THIRTEENTH BYTE	MAXIMUM OUTSTANDING CREDIT (LOW-ORDER BITS)	

# FIG.45

## PRIOR ART

HEADER	FIRST BYTE	PSID	CONTENTS : 0x00
	SECOND BYTE	SSID	CONTENTS : 0x00
	THIRD BYTE	LENGTH (NUMBER OF DATA, HIGH-ORDER BITS)	CONTENTS : 0x000C (WITH 2 BYTES)
	FOURTH BYTE	LENGTH (NUMBER OF DATA, LOW-ORDER BITS)	
	FIFTH BYTE	CREDIT	CONTENTS : 0x00~0xFF
	SIXTH BYTE	CONTROL	CONTENTS : 0x00
	SEVENTH BYTE	COMMAND	CONTENTS : 0x84
	EIGHTH BYTE	RESULT	CONTENTS : 0x00 (IN NORMAL STATE)
	NINTH BYTE	PRIMARY SOCKET ID	CONTENTS : 0x01~0xFF
	TENTH BYTE	SECONDARY SOCKET ID	CONTENTS : 0x01~0xFF
	ELEVENTH BYTE	CREDIT (HIGH-ORDER BITS)	CONTENTS : 0x0000 ~0xFFFF
	TWELFTH BYTE	CREDIT (LOW-ORDER BITS)	